

| | | | | |
|---------------|-----------------|---------------|---------|---------------|
| RRRRRRRRRRRRR | TTTTTTTTTTTTTTT | PPPPPPPPPPPPP | AAAAAAA | DDDDDDDDDDDDD |
| RRRRRRRRRRRRR | TTTTTTTTTTTTTTT | PPPPPPPPPPPPP | AAAAAAA | DDDDDDDDDDDDD |
| RRRRRRRRRRRRR | TTTTTTTTTTTTTTT | PPPPPPPPPPPPP | AAAAAAA | DDDDDDDDDDDDD |
| RRR FRR | TTT | PPP | PPP AAA | DDD AAA |
| RRR RRR | TTT | PPP | PPP AAA | DDD AAA |
| RRR RRR | TTT | PPP | PPP AAA | DDD AAA |
| RRR RRR | TTT | PPP | PPP AAA | DDD AAA |
| RRR RRR | TTT | PPP | PPP AAA | DDD AAA |
| RRR RRR | TTT | PPP | PPP AAA | DDD AAA |
| RRR RRR | TTT | PPP | PPP AAA | DDD AAA |
| RRR RRR | TTT | PPP | PPP AAA | DDD AAA |
| RRR RRR | TTT | PPP | PPP AAA | DDD AAA |
| RRRRRRRRRRRRR | TTT | PPPPPPPPPPPPP | AAA | AAA DDD |
| RRRRRRRRRRRRR | TTT | PPPPPPPPPPPPP | AAA | AAA DDD |
| RRRRRRRRRRRRR | TTT | PPPPPPPPPPPPP | AAA | AAA DDD |
| RRR RRR | TTT | PPP | AAAAAAA | DDD |
| RRR RRR | TTT | PPP | AAAAAAA | DDD |
| RRR RRR | TTT | PPP | AAAAAAA | DDD |
| RRR RRR | TTT | PPP | AAAAAAA | DDD |
| RRR RRR | TTT | PPP | AAA | AAA DDD |
| RRR RRR | TTT | PPP | AAA | AAA DDD |
| RRR RRR | TTT | PPP | AAA | AAA DDD |
| RRR RRR | TTT | PPP | AAA | AAA DDD |

FILEID**RSXRT

E 2

RS)
VOL

The image displays a massive grid of black text symbols on a white background. The symbols are organized into several distinct clusters. In the top-left, there are two groups of 'RR' symbols. To their right is a cluster of 'SS' symbols, which is further divided into two sections by a diagonal line of 'XX' symbols. Above these are two groups of 'RRRRRRRR' symbols. To the right of the 'SS' cluster is a vertical column of 'TT' symbols. Below the 'RR' and 'SS' clusters is a large group of 'RRRRRRRR' symbols. To the right of this group is another vertical column of 'TT' symbols. In the center, there is a large, irregular cluster of 'SS' symbols. To the left of this central cluster is a vertical column of 'LL' symbols. To the right of the central 'SS' cluster is a vertical column of 'RR' symbols. At the bottom, there are two more groups of 'LL' symbols, one on the far left and one on the far right. Between these bottom 'LL' groups and the central 'SS' cluster is a vertical column of 'II' symbols.

```
1 0001 0 MODULE RSXRT (
2 0002 0 IDENT = 'V04-000'.
3 0003 0 ADDRESSING_MODE(INTERNAL=GENERAL)
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1 !
7 0007 1 !*****+
8 0008 1 !*
9 0009 1 !* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 !* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 !* ALL RIGHTS RESERVED.
12 0012 1 !*
13 0013 1 !* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
14 0014 1 !* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
15 0015 1 !* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
16 0016 1 !* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
17 0017 1 !* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
18 0018 1 !* TRANSFERRED.
19 0019 1 !*
20 0020 1 !* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
21 0021 1 !* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
22 0022 1 !* CORPORATION.
23 0023 1 !*
24 0024 1 !* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
25 0025 1 !* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
26 0026 1 !*
27 0027 1 !*
28 0028 1 !*****+
29 0029 1 !*
30 0030 1 !++
31 0031 1 !*
32 0032 1 !* FACILITY: REMOTE TERMINAL SUPPORT
33 0033 1 !*
34 0034 1 !* ABSTRACT:
35 0035 1 !* THIS PROGRAM SUPPORTS THE RSX-11M REMOTE TERMINAL PROTOCOL.
36 0036 1 !*
37 0037 1 !*
38 0038 1 !* ENVIRONMENT:
39 0039 1 !*
40 0040 1 !* VAX/VMS Operating System
41 0041 1 !*
42 0042 1 !-
43 0043 1 !*
44 0044 1 !*
45 0045 1 !* AUTHOR: W M CARDOZA, CREATION DATE: 2-JAN-80
46 0046 1 !*
47 0047 1 !* MODIFIED BY:
48 0048 1 !*
49 0049 1 !* V03-003 WMC0002 Wayne Cardoza 28-Feb-1984
50 0050 1 !* Fix check for cancel-all.
51 0051 1 !*
52 0052 1 !* V03-002 MHB0081 Mark Bramhall 1-Sep-1982
53 0053 1 !* Use IOS_TTYREADALL instead of IOS_READPBLK.
54 0054 1 !*
55 0055 1 !* V03-001 WMC0001 Wayne Cardoza 6-May-1982
56 0056 1 !* Check for valid CURRENTIO in CANCEL.
57 0057 1 !*
```

```
58 0058 1 !**
59 0059 1 LIBRARY 'SYSSLIBRARY:LIB';
60 0060 1 LIBRARY 'SYSSLIBRARY:CLIMAC';
61 0061 1 !
62 0062 1 !
63 0063 1 !
64 0064 1 FORWARD ROUTINE
65 0065 1 GETTERMCHAR: NOVALUE,
66 0066 1 GETBUF,
67 0067 1 FREEBUF,
68 0068 1 INDREAD,
69 0069 1 LINKRECV: NOVALUE,
70 0070 1 WRITE: NOVALUE,
71 0071 1 TERMMBXMSG: NOVALUE,
72 0072 1 READ: NOVALUE,
73 0073 1 CNTRLCAST: NOVALUE,
74 0074 1 CNTRLYAST: NOVALUE,
75 0075 1 READSINGLE: NOVALUE,
76 0076 1 ATTACH: NOVALUE,
77 0077 1 RSXRT: NOVALUE,
78 0078 1 LINKMBXMSG: NOVALUE,
79 0079 1 BROADCAST: NOVALUE,
80 0080 1 READPROMPT: NOVALUE,
81 0081 1 QIODEONE: NOVALUE,
82 0082 1 CANCEL: NOVALUE,
83 0083 1 TERMINATOR,
84 0084 1 UNSUPPORTED: NOVALUE,
85 0085 1 MAPMODIFIER,
86 0086 1 LINKWRTDONE: NOVALUE,
87 0087 1 NEXTIO: NOVALUE,
88 0088 1 UNSDATEENBL: NOVALUE,
89 0089 1 ONECHAR: NOVALUE;
90 0090 1 !
91 0091 1 !
92 0092 1 MACRO
93 0093 1 RTP_BUF = BLOCK[32] FIELD(RTP_FIELDS) %,
94 0094 1 QUIT = BEGIN
95 0095 1     SSETAST (ENBFLG = 0); ! STOP EVERYTHING
96 0096 1     WAKEFLAG = 1;
97 0097 1     SWAKE(); ! WAKE UP BASE LEVEL
98 0098 1     RETURN;
99 0099 1     END %.
100 0100 1 QUIT_ON_ERROR = IF (.RETCODE AND 1) EQL 0 THEN
101 0101 1     QUIT %;
102 0102 1 !
103 0103 1 ! EQUATED SYMBOLS:
104 0104 1 !
105 0105 1 LITERAL
106 0106 1 ! FUNCTION CODES
107 0107 1 RF_NOP = 0.           ! NOP
108 0108 1 RF_SSD = 1.          ! CONFIGURATION
109 0109 1 RF_DIS = 2.          ! DISCONNECT
110 0110 1 RF_WTD = 3.          ! WRITE DATA
111 0111 1 RF_RDD = 4.          ! READ DATA
112 0112 1 RF_WRD = 5.          ! READ WITH PROMPT
113 0113 1 RF_UNA = 6.          ! UNSOLICITED INPUT DISABLE/ENABLE
114 0114 1 RF_RSC = 7.          ! READ SINGLE CHARACTERS
```

```

115      0115 1      RF_KIL = 8.          | CANCEL I/O
116      0116 1      RF_ATT = 9.        | ATTACH
117      0117 1      RF_GTC = 10.       | GET TERMINAL CHARACTERISTICS
118      0118 1      RF_STC = 11.       | SET TERMINAL CHARACTERISTICS
119      0119 1      RF_ECR = 12.       | EXCEPTION CONDITION
120      0120 1      ! MODIFIERS
121      0121 1      RM_WBN = 1.        | WRITE BINARY
122      0122 1      RM_WBT = 2.        | BROADCAST
123      0123 1      RM_RBN = 4.        | READ BINARY
124      0124 1      RM_RTC = 8.        | READ TERMINATES ON CONTROL CHARACTERS
125      0125 1      RM_RNE = 16.       | READ NO ECHO
126      0126 1      RM_RTO = 32.       | RESET TIME OUT ON EACH CHARACTER
127      0127 1      RM_DET = 128.      | DETACH TERMINAL
128      0128 1      RM_NWC = 128.     | NO WRITE COMPLETE STATUS
129      0129 1      RM_TUI = 128.     | TERMINATE UNSOLICITED INPUT
130      0130 1      RM_TSC = 128.     | TERMINATE SINGLE CHARACTER INPUT
131      0131 1      ! FLAGS
132      0132 1      RM_PRI = 2.        | PROCESS REQUEST IMMEDIATELY
133      0133 1      RM_CAO = 4.        | CANCEL ABORT OUTPUT
134      0134 1      ! STATUS CODES
135      0135 1      RS_SFC = 0.        | SUCCESS
136      0136 1      RS_FPE = 1.        | FUNCTION PROCESSING ERROR
137      0137 1      RS_UFC = 2.        | UNSUPPORTED FUNCTION
138      0138 1      RS_IPF = 3.        | ILLEGAL PROTOCOL FUNCTION
139      0139 1      RS_IPD = 4.        | ILLEGAL PROTOCOL DATA
140      0140 1      RS_ICF = 5.        | ILLEGAL CHARACTERISTICS FUNCTION
141      0141 1      ! TERMINAL CHARACTERISTIC CODES
142      0142 1      RC_MHT = 18.       | HARDWARE TABS
143      0143 1      RC_NEA = 19.       | NO ECHO
144      0144 1      RC_TTF = 22.       | TERMINAL TYPE
145      0145 1      RC_SCP = 23.       | CRT
146      0146 1      RC_BIN = 24.       | BINARY MODE
147      0147 1      RC_TPL = 28.       | PAGE LENGTH
148      0148 1      RC_MAX = 28.       | ***** KEEP THIS THE MAXIMUM *****
149      0149 1      ! EXCEPTION CONDITION CODES
150      0150 1      RE_SAR = 0;        | SYSTEM ATTENTION REQUEST
151      0151 1      FIELD
152      0152 1      RTP_FIELDS =
153      0153 1      ! REMOTE TERMINAL PROTOCOL
154      0154 1      SET
155      0155 1      RTP_LNK = [0,0,32,0].   | QUEUE LINK WORDS
156      0156 1      RTP_LN2 = [1,0,32,0].
157      0157 1      RTP_IOS = [2,0,16,0].
158      0158 1      RTP_IOC = [2,16,16,0]. | IOSB
159      0159 1      RTP_IO2 = [3,0,32,0]. | I/O COUNT
160      0160 1      RTP_FNC = [4,0,8,0].  | FUNCTION CODE
161      0161 1      RTP_MOD = [4,8,8,0].  | FUNCTION MODIFIER BITS
162      0162 1      RTP_FLG = [4,16,8,0]. | FUNCTION FLAGS
163      0163 1      RTP_STS = [4,24,8,0]. | RETURN STATUS
164      0164 1      RTP_IDN = [5,0,8,0].  | IDENTIFIER
165      0165 1      RTP_RSV = [5,8,8,0].  | RESERVED, MBZ
166      0166 1      RTP_RCT = [5,16,16,0]. | RECEIVE BYTE COUNT
167      0167 1      RTP_TCT = [6,0,16,0]. | TRANSMIT BYTE COUNT
168      0168 1      RTP_DAT = [6,16,32,0]. | DATA
169      0169 1      TES;
170      0170 1      !
171      0171 1      !

```

172 0172 1 OWN
173 0173 1 NAMEIOSB:
174 0174 1 VMSCONFIG:
175 0175 1 VECTOR[4,WORD],
176 0176 1 INITIAL(PLT BYTE(RF_SSD,1,0,0,
177 0177 1 WORD(4,2),
178 0178 1 WORD(128),
179 0179 1 2,1,
180 0180 1 3,1,
181 0181 1 5,1,
182 0182 1 7,1,
183 0183 1 8,1,
184 0184 1 9,1,
185 0185 1 10,1,
186 0186 1 11,1,
187 0187 1 12,1,
188 0188 1 13,1,
189 0189 1 127,1,
190 0190 1 0,0))!
191 0191 1 TERMMBXDATA:
192 0192 1 UNSOLENBLFLG:
193 0193 1 ATTACHFLAG:
194 0194 1 SINGLEINPROG:
195 0195 1 UNSOLPEND:
196 0196 1 READINPROG:
197 0197 1 SINGLEFLAG:
198 0198 1 CURRENTIO:
199 0199 1 INDDATA:
200 0200 1 IOQUEUE:
201 0201 1 IOQUEUE:
202 0202 1 CNTRLCMMSG:
203 0203 1 LINKMAIL:
204 0204 1 STERMASK:
205 0205 1 STERMDesc:
206 0206 1 NTERMDESC:
207 0207 1 NTERMDesc:
208 0208 1 REQ_DSCNTRLY:
209 0209 1 ! THIS TELLS REMOTE TERMINAL MAIN PROGRAM WHAT PROTOCOL WE SUPPORT
210 0210 1 PSECT OWN = PROTOTBL (ALIGN(0));
211 0211 1 OWN
212 0212 1 PROTOMASK: WORD INITIAL(2),
213 0213 1 RSXADDR: ALIGN(0) INITIAL(RSXRT); ! RSX-11
214 0214 1 ! EXTERNAL REFERENCES:
215 0215 1 EXTERNAL ROUTINE
216 0216 1 SYSSCLI : ADDRESSING_MODE(LONG_RELATIVE),
217 0217 1 LIB\$GET_VM;
218 0218 1 BUILTIN
219 0219 1 INSQUE,
220 0220 1 REMQUE;
221 0221 1 EXTERNAL
222 0222 1 TTYDESC,
223 0223 1 KEMS_NEFDIS,
224 0224 1 RDWRTCHAN: WORD,
225 0225 1 CNTRLCHAN: WORD,
226 0226 1 TERMMBXCHAN: WORD,

J 2
16-Sep-1984 02:18:51
14-Sep-1984 13:04:57

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RTPAD.SRC]RSXRT.B32;1

Page 5
(1)

: 229 0229 1 MAILCHAN: WORD,
: 230 0230 1 LINKCHAN: WORD,
: 231 0231 1 SYSINRAB: \$RAB_DECL,
: 232 0232 1 SYSINFAB: \$FAB_DECL,
: 233 0233 1 INDFLAG: BYTE,
: 234 0234 1 WAKEFLAG: BYTE,
: 235 0235 1 RETSTATUS;

```
0237    0236 1 ROUTINE RSXRT: NOVALUE =
0238    0237 1 !++
0239    0238 1
0240    0239 1 Functional Description:
0241    0240 1     Performs initialization functions for RSX remote terminals.
0242    0241 1
0243    0242 1
0244    0243 1 Calling Sequence:
0245    0244 1     standard
0246    0245 1
0247    0246 1 Input Parameters:
0248    0247 1     none
0249    0248 1
0250    0249 1 Implicit Inputs:
0251    0250 1     none
0252    0251 1
0253    0252 1 Output Parameters:
0254    0253 1     none
0255    0254 1
0256    0255 1 Implicit Outputs:
0257    0256 1     none
0258    0257 1
0259    0258 1 Routines Called:
0260    0259 1     GETBUF
0261    0260 1
0262    0261 1 Routine Value:
0263    0262 1     none
0264    0263 1
0265    0264 1 Signals:
0266    0265 1     none
0267    0266 1
0268    0267 1 Side Effects:
0269    0268 1     A configuration message is transmitted.
0270    0269 1     A prompt is displayed on the screen.
0271    0270 1     Reads are initiated on the terminal mailbox and on the link.
0272    0271 1
0273    0272 1 --.
0274    0273 2 BEGIN
0275    0274 2 LOCAL
0276    0275 2     BUFFER: REF RTP_BUF;
0277    0276 2     RETSTATUS =
0278    P 0277 2     SQIOW (CHAN = .LINKCHAN,      ! SEND CONFIGURATION MESSAGE
0279    P 0278 2             FUNC = IOS_WRITEVBLK,
0280    P 0279 2             P1 = .VMS$CONFIG,
0281    P 0280 2             P2 = 4 * (.VMS$CONFIG-4));
0282    0281 2     QUIT_ON_ERROR;
0283    0282 2     RETSTATUS =
0284    P 0283 2     SQIOW (CHAN = .RDWRCHAN,      ! ENABLE UNSOLICITED INPUT
0285    0284 2             FUNC = IOS_WRITEVBLK+IOS$M_ENABLMBX);
0286    0285 2     QUIT_ON_ERROR;
0287    0286 2     RETSTATUS =
0288    P 0287 2     SQIO  (CHAN = .TERMMBXCHAN,      ! UNSOLICITED DATA MBX READ
0289    P 0288 2             FUNC = IOS_READVBLK,
0290    P 0289 2             ASTADR = TERMMBXMSG,
0291    P 0290 2             P1 = TERMMBXDATA,
0292    P 0291 2             P2 = 8);
0293    0292 2     QUIT_ON_ERROR;
```

L 2
16-Sep-1984 02:18:51
14-Sep-1984 13:04:57VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RTPAD.SRC]RSXRT.B32:1Page 7
(2)

```

294      0293 2      RETSTATUS =
295      P 0294 2      $QIO   (CHAN = .MAILCHAN,
296      P 0295 2      FUNC = IOS_READVBLK,
297      P 0296 2      ASTADR = LINKMBXMSG,
298      P 0297 2      P1 = LINKMAIL,
299      P 0298 2      P2 = 40);
300      P 0299 2      QUIT_ON_ERROR;
301      P 0300 2      RETSTATUS =
302      P 0301 2      $QIO   (CHAN = .CNTRLCHAN, ! HANDLE CONTROL-C
303      P 0302 2      FUNC = IOS_SETMODE+IOSM_CTRLCAST,
304      P 0303 2      P1 = CNTRLCAST);
305      P 0304 2      QUIT_ON_ERROR;
306      P 0305 2      RETSTATUS =
307      P 0306 2      $QIO   (CHAN = .CNTRLCHAN, ! HANDLE CONTROL-Y
308      P 0307 2      FUNC = IOS_SETMODE+IOSM_CTRLYAST,
309      P 0308 2      P1 = CNTRLYAST);
310      P 0309 2      QUIT_ON_ERROR;
311      P 0310 2      SYSSCLITREQ_DSCNTRLY,0,0); ! DISABLE CLI ^Y
312      P 0311 2      RETSTATUS =
313      P 0312 2      $QIOW  (CHAN = .RDWRCHAN, ! GIVE AN RSX PROMPT
314      P 0313 2      FUNC = IOS_WRITEVBLK,
315      P 0314 2      P1 = UPLIT-BYTE('>'),
316      P 0315 2      P2 = 1);
317      P 0316 2      QUIT_ON_ERROR;
318      P 0317 2      IF .INDFLAG NEQ 0 THEN
319      P 0318 3      BEGIN
320      P 0319 3      INDDATA = GETBUF(); ! THERE IS AN INDIRECT FILE
321      P 0320 3      ! GET BUFFER FOR FILE READ
322      P 0321 3      SYSINRAB[RAB$L_UBF] = INDDATA[RTP-DAT]; ! BUFFER ADDRESS
323      P 0322 3      SYSINRAB[RAB$W_USZ] = 100; ! ALLOW 100 CHARACTERS
324      P 0323 2      INREAD(); ! READ IT
325      P 0324 2      END;
326      P 0325 2      BUFFER = GETBUF(); ! REQUEST A BUFFER
327      P 0326 2      RETSTATUS =
328      P 0327 2      $QIO   (CHAN = .LINKCHAN, ! WAIT FOR SOMETHING ON LINK
329      P 0328 2      FUNC = IOS_READVBLK,
330      P 0329 2      IOSB = BUFFER[RTP_IOS],
331      P 0330 2      ASTADR = LINKRECV,
332      P 0331 2      ASTPRM = .BUFFER,
333      P 0332 2      P1 = BUFFER[RTP_FNC],
334      P 0333 2      P2 = 128);
335      P 0334 1      QUIT_ON_ERROR;
END;

```

```

.TITLE RSXRT
.IDENT \V04-000\
.PSECT PROTOTBL,NOEXE,0

```

```

0002 00000 PROTOMASK:
00000000' 00002 RSXADDR:.ADDRESS RSXRT
.PSECT SPLITS,NOWRT,NOEXE,2
00 00 0000009 00000 .LONG 9
00 00 01 01 00004 P.AAA: .BYTE 1, 1, 0, 0

```

0B 01 0A 01 09 01 08 01 07 01 05 01 03 C1 02 0000E .WORD 4 2
00 00 01 0D 01 0C 01 0001D .WORD 128
00026 .BYTE 2, 1, 3, 1, 5, 1, 7, 1, 8, 1, 9, 1, 10, -
3E 00028 P.AAB: .BLKB 2
.PSECT \$OWNS,NOEXE,2

00000 NAMEIOSB:
00000000' 00008 VMSCONFIG: .BLKB 8
00000000 0000C TERMMBXDATA: .ADDRESS P.AAA
00000000 00014 UNSOLENBLFLG: .BLKB 8
00 00018 ATTACHFLAG: .LONG 0
00 00019 SINGLEINPROG: .BYTE 0
00 0001A UNSOLPEND: .BYTE 0
00 0001B READINPROG: .BYTE 0
00000000 0001C SINGLEFLAG: .BYTE 0
00000000 00020 CURRENTIO: .LONG 0
00000000 00024 INDDATA: .LONG 0
00000000' 00028 IOQUEUE: .ADDRESS IOQUEUE, IOQUEUE
00000000' 00030 BUFQUEUE: .ADDRESS BUFQUEUE, BUFQUEUE
00 00 00 0C 00038 CNTRLCMSG: .BYTE 12, 0, 0, 0
0003C LINKMAIL: .BLKB 40
E0000000 00000000 00000000 FFFFFFFF 00064 STERMASK: .LONG -1, 0, 0, -536870912
00000010 00074 STERMDESC: .LONG 16
00000000' 00078 .ADDRESS STERMASK
0C002000 0007C NTERMMASK: .LONG 201334784
00000004 00080 NTERMDESC: .LONG 4
00000000' 00084 .ADDRESS NTERMMASK
05 00088 REQ_DSCNTRLY: .BYTE 5
08 00089 .BYTE 8
00 0008A .BYTE 0
00 0008B .BYTE 0
00000000 0008C .LONG 0
00000000 00090 .LONG 0
00000000 00098 .LONG 0
00000000 0009C .LONG 0
00000000 000A0 .LONG 0

| | | | | | | |
|--|--|--|--|---|---------------------------|------|
| | | | | .EXTRN SYSSCLI, LIB\$GET VM | | |
| | | | | .EXTRN TTYDESC, REMS NETDIS | | |
| | | | | .EXTRN RDWRCHAN, CNTRLCHAN | | |
| | | | | .EXTRN TERMMBXCHAN, MAILCHAN | | |
| | | | | .EXTRN LINKCHAN, SYSINRAB | | |
| | | | | .EXTRN SYSINFAB, INDFLAG | | |
| | | | | .EXTRN WAKEFLAG, RETSTATUS | | |
| | | | | .EXTRN SYSSQIOW, SYSSSETAST | | |
| | | | | .EXTRN SYSSWAKE, SYSSQIO | | |
| | | | | .PSECT SCODES,NOWRT,2 | | |
| | | | | | | |
| | | | | 58 00000000G 00 9E 00000 RSXRT: .WORD Save R2,R3,R4,R5,R6,R7,R8 | | 0236 |
| | | | | 57 00000000G 00 9E 00009 | MOVAB CNTRLCHAN, R8 | |
| | | | | 56 00000000G 00 9E 00010 | MOVAB RDWRCHAN, R7 | |
| | | | | 55 0000' CF 9E 00017 | MOVAB LINKCHAN, R6 | |
| | | | | 54 00000000G 00 9E 0001C | MOVAB INDDATA, R5 | |
| | | | | 53 00000000G 00 9E 00023 | MOVAB SYSSQIOW, R4 | |
| | | | | 52 00000000G 00 9E 0002A | MOVAB SYSSQIO, R3 | |
| | | | | 7E 7C 00031 | MOVAB RETSTATUS, R2 | |
| | | | | 7E 7C 00033 | CLRQ -(SP) | |
| | | | | A5 D0 00035 | CLRQ -(SP) | |
| | | | | 02 78 00039 | MOVL VMSCONFIG, R0 | |
| | | | | 50 DD 0003E | ASHL #2, -4(R0), -(SP) | |
| | | | | 7E 7C 00040 | PUSHL R0 | |
| | | | | 30 7D 00042 | CLRQ -(SP) | |
| | | | | 66 3C 00045 | MOVQ #48, -(SP) | |
| | | | | 7E D4 00048 | MOVZWL LINKCHAN, -(SP) | |
| | | | | OC FB 0004A | CLRL -(SP) | |
| | | | | 50 D0 0004D | CALLS #12, SYSSQIOW | |
| | | | | 62 E9 00050 | MOVL R0, RETSTATUS | |
| | | | | 7E 7C 00053 | BLBC RETSTATUS, 1\$ | |
| | | | | 7E 7C 00055 | CLRQ -(SP) | |
| | | | | 7E 7C 00057 | CLRQ -(SP) | |
| | | | | 7E 7C 00059 | CLRQ -(SP) | |
| | | | | 7E D4 0005B | CLRL -(SP) | |
| | | | | 8F 9A 0005D | MOVZBL #176, -(SP) | |
| | | | | 67 3C 00061 | MOVZWL RDWRCHAN, -(SP) | |
| | | | | 7E D4 00064 | CLRL -(SP) | |
| | | | | OC FB 00066 | CALLS #12, SYSSQIOW | |
| | | | | 50 D0 00069 | MOVL R0, RETSTATUS | |
| | | | | 62 E9 0006C | BLBC RETSTATUS, 2\$ | |
| | | | | 7E 7C 0006F | CLRQ -(SP) | |
| | | | | 7E 7C 00071 | CLRL -(SP) | |
| | | | | 08 DD 00073 | PUSHL #8 | |
| | | | | A5 9F 00075 | PUSHAB TERMMBXDATA | |
| | | | | 7E D4 00078 | CLRL -(SP) | |
| | | | | CF 9F 0007A | PUSHAB TERMMBXMSG | |
| | | | | 31 7D 0007E | MOVL #49, -(SP) | |
| | | | | 00 3C 00081 | MOVZWL TERMMBXCHAN, -(SP) | |
| | | | | 7E D4 00088 | CLRL -(SP) | |
| | | | | OC FB 0008A | CALLS #12, SYSSQIO | |
| | | | | 50 D0 0008D | MOVL R0, RETSTATUS | |
| | | | | 62 E9 00090 | BLBC RETSTATUS, 3\$ | |
| | | | | 7E 7C 00093 | CLRQ -(SP) | |

| | | | | | | | |
|--|-----------|----|----|----------|--------|-----------------|--------|
| | | 7E | 7E | 7C 0014B | CLRQ | -(SP) | : 0332 |
| | | 80 | 7E | 7C 0014D | CLRQ | -(SP) | |
| | | 10 | 8F | 9A 0014F | MOVZBL | #128, -(SP) | |
| | | | A0 | 9F 00153 | PUSHAB | 16(BUFFER) | |
| | 00000V | | 50 | DD 00156 | PUSHL | BUFFER | |
| | | 08 | CF | 9F 00158 | PUSHAB | LINKRECV | |
| | | | A0 | 9F 0015C | PUSHAB | 8(BUFFER) | |
| | | | 31 | DD 0015F | PUSHL | #49 | |
| | 7E | | 66 | 3C 00161 | MOVZWL | LINKCHAN, -(SP) | |
| | | | 7E | D4 00164 | CLRL | -(SP) | |
| | 63 | | 0C | FB 00166 | CALLS | #12, SYSSQIO | |
| | 62 | | 50 | DC 00169 | MOVL | R0, RETSTATUS | |
| | 19 | | 62 | E8 0016C | BLBS | RETSTATUS, 7\$ | |
| | 00000000G | 00 | 7E | D4 0016F | CLRL | -(SP) | |
| | 00000000G | 00 | 01 | FB 00171 | CALLS | #1, SYSSSETAST | |
| | 00000000G | 00 | 01 | 90 00178 | MOVB | #1, WAKEFLAG | |
| | | | 7E | 7C 0017F | CLRQ | -(SP) | |
| | | | 02 | FB 00181 | CALLS | #2, SYSSWAKE | |
| | | | 04 | 00188 | 7\$: | RET | |

; Routine Size: 393 bytes. Routine Base: \$CODE\$ + 0000

```
337 0335 1 ROUTINE GETBUF =
338 0336 1 ++
339 0337 1
340 0338 1 Functional Description:
341 0339 1 Allocate a buffer.
342 0340 1
343 0341 1
344 0342 1 Calling Sequence:
345 0343 1 standard
346 0344 1
347 0345 1 Input Parameters:
348 0346 1 none
349 0347 1
350 0348 1 Implicit Inputs:
351 0349 1 BUFQUEUE
352 0350 1
353 0351 1 Output Parameters:
354 0352 1 none
355 0353 1
356 0354 1 Implicit Outputs:
357 0355 1 none
358 0356 1
359 0357 1 Routines Called:
360 0358 1 LIB$GET_VM
361 0359 1
362 0360 1 Routine Value:
363 0361 1 buffer address
364 0362 1
365 0363 1 Signals:
366 0364 1 none
367 0365 1
368 0366 1 Side Effects:
369 0367 1 none
370 0368 1
371 0369 1 ---
372 0370 2 BEGIN
373 0371 2 LOCAL
374 0372 2 BUFADR:;
375 0373 2 IF REMQUE(.BUFQUEUE,BUFADR) EQL 3 THEN ! WAS QUEUE EMPTY?
376 0374 2 LIB$GET_VM(UPLIT(128+16),BUFADR); ! GET A BUFFER
377 0375 2 RETURN .BUFADR;
378 0376 1 END;
```

.PSECT \$SPLIT\$,NOWRT,NOEXE,2
00000090 00029 00000090 0002C P.AAC: .BLKB 3
.LONG 144

.PSECT \$CODE\$,NOWRT,2
7E 0000' 0000 000000 GETBUF: .WORD Save nothing
DF OF 00002 REMQUE @BUFQUEUE, BUFADR
50 DC 00007 MOVPSL R0

| | | | | |
|----------|----|-------------------|------------------------|----------------------|
| 50 | 50 | 02 | 01 EF 00009 | EXTIV #1, #2, R0, R0 |
| | | 03 | 50 D1 0000E | CMPL R0, #3 |
| | | | 0D 12 00011 | BNEQ 1\$ |
| | | | 5E DD 00013 | PUSHL SP |
| | | 0000' CF 9F 00015 | PUSHAB P.AAC | |
| 0000000G | 00 | 02 FB 00019 | CALI\$ #2, LIB\$GET_VM | |
| | 50 | 6E D0 00020 | MCVL BUFADR, R0 | |
| | | 1\$: 04 00023 | RET | |

: 0374
: 0375
: 0376

; Routine Size: 36 bytes, Routine Base: \$CODE\$ + 0189

; 379 0377 1

```
: 381      0378 1 ROUTINE FREEBUF(BUF) =
: 382      0379 1 ++
: 383      0380 1
: 384      0381 1 Functional Description:
: 385      0382 1 Release a buffer.
: 386      0383 1
: 387      0384 1
: 388      0385 1 Calling Sequence:
: 389      0386 1 standard
: 390      0387 1
: 391      0388 1 Input Parameters:
: 392      0389 1 BUF = buffer address
: 393      0390 1
: 394      0391 1 Implicit Inputs:
: 395      0392 1 none
: 396      0393 1
: 397      0394 1 Output Parameters:
: 398      0395 1 none
: 399      0396 1
: 400      0397 1 Implicit Outputs:
: 401      0398 1 BUFQUEUE
: 402      0399 1
: 403      0400 1 Routines Called:
: 404      0401 1 none
: 405      0402 1
: 406      0403 1 Routine Value:
: 407      0404 1 none
: 408      0405 1
: 409      0406 1 Signals:
: 410      0407 1 none
: 411      0408 1
: 412      0409 1 Side Effects:
: 413      0410 1 none
: 414      0411 1
: 415      0412 1 ---
: 416      0413 2 BEGIN
: 417      0414 2 INSQUE(.BUF,BUFQUEUE)
: 418      0415 1 END;
```

| | | | | |
|----------|----|-----------------------------------|--------------|--------|
| 0000' CF | 04 | 0000 00000 FREEBUF: .WORD | Save nothing | : 0378 |
| | | 50 D4 00002 CLRL R0 | | : 0414 |
| | | BC 0E 00004 INSQUE @BUF, BUFQUEUE | | |
| | | 02 12 0000A BNEQ 1\$ | | |
| | | 50 D6 0000C INCL R0 | | |
| | | 04 0000E 1\$: RET | | : 0415 |

; Routine Size: 15 bytes, Routine Base: \$CODE\$ + 01AD

```
: 420      0416 1 ROUTINE LINKRECV(BUFFER): NOVALUE =
: 421      0417 1 ++
: 422      0418 1
: 423      0419 1 Functional Description:
: 424      0420 1     Receive a message on the link and call the correct service routine.
: 425      0421 1
: 426      0422 1
: 427      0423 1 Calling Sequence:
: 428      0424 1     standard
: 429      0425 1
: 430      0426 1 Input Parameters:
: 431      0427 1     BUFFER = input buffer address
: 432      0428 1
: 433      0429 1 Implicit Inputs:
: 434      0430 1     none
: 435      0431 1
: 436      0432 1 Output Parameters:
: 437      0433 1     none
: 438      0434 1
: 439      0435 1 Implicit Outputs:
: 440      0436 1     RETSTATUS
: 441      0437 1
: 442      0438 1 Routines Called:
: 443      0439 1     WRITE
: 444      0440 1     READ
: 445      0441 1     READPROMPT
: 446      0442 1     UNSDATEENBL
: 447      0443 1     CANCEL
: 448      0444 1     READSINGLE
: 449      0445 1     ATTACH
: 450      0446 1     UNSUPPORTED
: 451      0447 1     GETBUF
: 452      0448 1     GETTERMCHAR
: 453      0449 1
: 454      0450 1 Routine Value:
: 455      0451 1     none
: 456      0452 1
: 457      0453 1 Signals:
: 458      0454 1     none
: 459      0455 1
: 460      0456 1 Side Effects:
: 461      0457 1     A new read to the link is initiated.
: 462      0458 1     If there is an error on the read, a SWAKE is issued to force the
: 463      0459 1     program to exit.
: 464      0460 1
: 465      0461 1 -- BEGIN
: 466      0462 2 LOCAL
: 467      0463 2
: 468      0464 2     NEWBUF: REF RTP_BUF;
: 469      0465 2     MAP BUFFER: REF RTP_BUF;
: 470      0466 2     RETSTATUS = .BUFFER[RTP_IOS];
: 471      0467 2     IF .RETSTATUS EQL SSS_ABORT THEN
: 472      0468 2     RETURN;           ! Link gone - mailbox message will tell why
: 473      0469 2     QUIT_ON_ERROR;
: 474      0470 2     CASE .BUFFER[RTP_FNC] FROM 0 TO 12 OF
: 475      0471 2     SET
: 476      0472 2     [RF_WTD]:      WRITE(.BUFFER);
```

```

477      0473 2
478      0474 2
479      0475 2
480      0476 2
481      0477 2
482      0478 2
483      0479 2
484      0480 2
485      0481 2
486      0482 2
487      0483 2
488      0484 2
489      P 0485 2
490      P 0486 2
491      P 0487 2
492      P 0488 2
493      P 0489 2
494      P 0490 2
495      P 0491 2
496      0492 2
497      0493 2
498      0494 2
499      0495 1

[RF_RDD]:    READ(.BUFFER);
[RF_WRD]:    READPROMPT(.BUFFER);
[RF_JNS]:    UNSDATENBL(.BUFFER);
[RF_IIL]:    CANCEL(.BUFFER);
[RF_RSC]:    READSINGLE(.BUFFER);
[RF_ATT]:    ATTACH(.BUFFER);
[RF_GTC]:    GETTERMCHAR(.BUFFER);
[INRANGE]:   UNSUPPORTED(.BUFFER);
[OUTRANGE]:  UNSUPPORTED(.BUFFER);
TES:
NEWBUF = GETBUF();           ! GET ANOTHER BUFFER
RETSTATUS =
$QIO  (CHAN = .LINKCHAN,          ! READ LINK AGAIN
       FUNC = IOS_READVBLK,
       IOSB = NEWBUF[RTP_IOS],
       ASTADR = LINKRECV,
       ASTPRM = .NEWBUF,
       P1 = NEWBUF[RTP_FNC],
       P2 = 128);
IF .RETSTATUS EQL $SSS_ABORT THEN
  RETURN;                      ! Link gone - mailbox msg will tell why
  QUIT_ON_ERROR;
END;

```

| 000C 00000 LINKRECV: | | | | | | | | | |
|----------------------|-----------|----|------|-------|--------|------------|-----------|-----------------|------|
| 53 | 00000000G | 00 | 9E | 00002 | .WORD | Save R2,R3 | | | 0416 |
| 52 | 04 | AC | D0 | 00009 | MOVAB | RETSTATUS | R3 | | 0466 |
| 63 | 08 | A2 | 3C | 0000D | MOVL | BUFFER, | R2 | | 0467 |
| 50 | | 63 | D0 | 00011 | MOVZWL | 8(R2), | RETSTATUS | | |
| 2C | | 50 | D1 | 00014 | MOVL | RETSTATUS, | R0 | | |
| | | 01 | 12 | 00017 | CMPL | R0, | #44 | | |
| | | | 04 | 00019 | BNEQ | 1\$ | | | |
| | | | | | RET | | | | |
| 03 | | 50 | E8 | 0001A | 1\$: | BLBS | R0 | 2\$ | 0468 |
| 0C | 00 | 10 | 00A6 | 31 | 0001D | BRW | 14\$ | | 0470 |
| 0064 | 0064 | | A2 | 8F | 00020 | 2\$: | CASEB | 16(R2), #0, #12 | |
| 0049 | 0037 | | 0064 | | 00025 | 3\$: | .WORD | 12\$-3\$,- | |
| 0064 | 0058 | | 002E | | 0002D | | | 12\$-3\$,- | |
| | | | 0025 | | 00035 | | | 12\$-3\$,- | |
| | | | 0040 | | 00035 | | | 4\$-3\$,- | |
| | | | 0064 | | 0003D | | | 5\$-3\$,- | |
| | | | | | | | | 6\$-3\$,- | |
| | | | | | | | | 7\$-3\$,- | |
| | | | | | | | | 9\$-3\$,- | |
| | | | | | | | | 8\$-3\$,- | |
| | | | | | | | | 10\$-3\$,- | |
| | | | | | | | | 11\$-3\$,- | |
| | | | | | | | | 12\$-3\$,- | |
| | | | | | | | | 12\$-3\$, | |
| 0000V CF | | 48 | 11 | 0003F | 4\$: | BRB | 12\$ | | 0481 |
| | | 52 | DD | 00041 | | PUSHL | R2 | | 0472 |
| | | 01 | FB | 00043 | | CALLS | #1 | WRITE | |
| | | 46 | 11 | 00048 | | BRB | 13\$ | | |

| | | | |
|--------------|----------------------|------------------------|--------|
| 0000V CF | 52 DD 0004A 5\$: | PUSHL R2 | : 0473 |
| | 01 FB 0004C | CALLS #1 | |
| | 3D 11 00051 | BRB 13\$ READ | |
| 0000V CF | 52 DD 00053 6\$: | PUSHL R2 | : 0474 |
| | 01 FB 00055 | CALLS #1 | |
| | 34 11 0005A | BRB 13\$ READPROMPT | |
| 0000V CF | 52 DD 0005C 7\$: | PUSHL R2 | : 0475 |
| | 01 FB 0005E | CALLS #1 | |
| | 2B 11 00063 | BRB 13\$ UNSDATEENBL | |
| 0000V CF | 52 DD 00065 8\$: | PUSHL R2 | : 0476 |
| | 01 FB 00067 | CALLS #1 | |
| | 22 11 0006C | BRB 13\$ CANCEL | |
| 0000V CF | 52 DD 0006E 9\$: | PUSHL R2 | : 0477 |
| | 01 FB 00070 | CALLS #1 | |
| | 19 11 00075 | BRB 13\$ READSINGLE | |
| 0000V CF | 52 DD 00077 10\$: | PUSHL R2 | : 0478 |
| | 01 FB 00079 | CALLS #1 | |
| | 10 11 0007E | BRB 13\$ ATTACH | |
| 0000V CF | 52 DD 00080 11\$: | PUSHL R2 | : 0479 |
| | 01 FB 00082 | CALLS #1 | |
| | 07 11 00087 | BRB 13\$ GETTERMCHAR | |
| 0000V CF | 52 DD 00089 12\$: | PUSHL R2 | : 0480 |
| | 01 FB 0008B | CALLS #1, UNSUPPORTED | |
| FF38 CF | 00 FB 00090 13\$: | CALLS #0, GETBUF | : 0483 |
| | 7E 7C 00095 | CLRQ -(SP) | : 0491 |
| | 7E 7C 00097 | CLRQ -(SP) | |
| 7E | 80 8F 9A 00099 | MOVZBL #128, -(SP) | |
| | 10 A0 9F 0009D | PUSHAB 16(NEWBUF) | |
| | 50 DD 000A0 | PUSHL NEWBUF | |
| | FF5A CF 9F 000A2 | PUSHAB LINKRECV | |
| | 08 A0 9F 000A6 | PUSHAB 8(NEWBUF) | |
| | 31 DD 000A9 | PUSHL #49 | |
| 7E 00000000G | 00 3C 000AB | MOVZWL LINKCHAN, -(SP) | |
| | 7E D4 000B2 | CLRL -(SP) | |
| 00000000G | 00 0C FB 000B4 | CALLS #12, SYSSQIO | |
| | 63 50 D0 000BB | MOVL R0, RETSTATUS | |
| | 2C 50 D1 000BE | CMPL R0, #44 | : 0492 |
| | 1C 13 000C1 | BEQL 15\$ | |
| | 19 50 E8 000C3 | BLBS R0, 15\$ | : 0493 |
| 00000000G | 00 7E D4 000C6 14\$: | CLRL -(SP) | |
| 00000000G | 00 01 FB 000C8 | CALLS #1, SYSSSETAST | |
| | 01 90 000CF | MOVB #1, WAKEFLAG | |
| 00000000G | 00 7E 7C 000D6 | CLRQ -(SP) | |
| | 02 FB 000D8 | CALLS #2, SYSSWAKE | |
| | 04 000DF 15\$: | RET | : 0495 |

: Routine Size: 224 bytes, Routine Base: \$CODE\$ + 01BC

```
501      0496 1 ROUTINE WRITE(BUFFER): NOVALUE =
502      0497 1 ++
503      0498 1
504      0499 1 Functional Description:
505      0500 1 Perform a write QIO function to the terminal.
506      0501 1
507      0502 1 Calling Sequence:
508      0503 1 standard
509      0504 1
510      0505 1 Input Parameters:
511      0506 1 BUFFER = address of buffer from link
512      0507 1
513      0508 1 Implicit Inputs:
514      0509 1 CURRENTIO
515      0510 1
516      0511 1 Output Parameters:
517      0512 1 none
518      0513 1
519      0514 1 Implicit Outputs:
520      0515 1 IOQUEUE
521      0516 1
522      0517 1 Routines Called:
523      0518 1 BROADCAST
524      0519 1
525      0520 1 Routine Value:
526      0521 1 none
527      0522 1
528      0523 1 Signals:
529      0524 1 none
530      0525 1
531      0526 1 Side Effects:
532      0527 1 An I/O may be queued for later action
533      0528 1
534      0529 1 --
535      0530 2
536      0531 2 BEGIN
537      0532 2 MAP BUFFER: REF RTP BUF;
538      0533 2 IF (.BUFFER[RTP_MOD] AND RM_WBT) NEQ 0 THEN
539      0534 2     BROADCAST(.BUFFER) ! IT IS A BROADCAST WRITE
540      0535 3 ELSE
541      0536 3     BEGIN
542      0537 4         IF .CURRENTIO EQ 0 THEN
543      0538 4             BEGIN
544      P 0539 4                 RETSTATUS =
545      P 0540 4                 $QIO (CHAN = .RDWRCHAN, ! WRITE TO THE TERMINAL
546      P 0541 4                 FUNC = IOS_WRITEVBLK,
547      P 0542 4                 IOSB = BUFFER[RTP_IOS],
548      P 0543 4                 ASTADR = QIODENE,
549      P 0544 4                 ASTPRM = .BUFFER,
550      0545 4                 P1 = BUFFER[RTP_DAT],
551      0546 4                 P2 = BUFFER[RTP_TCT];
552      0547 4                 QUIT ON ERROR;
553      0548 4                 CURRENTIO = .BUFFER;
554      0549 3                 END
555      0550 3                 ELSE
556      0551 2                     INSQUE(.BUFFER,.IOQUEUE[1]); ! QUEUE IT FOR LATER
557      0552 1                 END;
```

| | | | | | | | | | | | | | |
|----------|----------|-------------|-------------|------------|--------|--------|------------------|--|--|--|--|--|------|
| | | | | | | | | | | | | | |
| 08 | 11 | 53 0000000G | 00 9E 00002 | 000C 00000 | WRITE: | .WORD | Save R2,R3 | | | | | | 0496 |
| | | 52 04 | AC D0 00009 | | | MOVAB | RETSTATUS, R3 | | | | | | 0532 |
| | | | 01 E1 0000D | | | MOVL | BUFFER, R2 | | | | | | 0533 |
| | | | 52 DD 00012 | | | BBC | #1, 17(R2), 1\$ | | | | | | 0536 |
| | | 0000V CF | 01 FB 00014 | | | PUSHL | R2 | | | | | | 0545 |
| | | | 04 00019 | | | CALLS | #1, BROADCAST | | | | | | 0547 |
| | | | | | | RET | | | | | | | 0548 |
| | | 0000' CF | D5 0001A | 1\$: | | TSTL | CURRENTIO | | | | | | 0549 |
| | | | 4C 12 C001E | | | BNEQ | 3\$ | | | | | | 0550 |
| | | | 7E 7C 00020 | | | CLRQ | -(SP) | | | | | | 0551 |
| | | | 7E 7C 00022 | | | CLRQ | -(SP) | | | | | | 0552 |
| | 7E | 18 | A2 3C 00024 | | | MOVZWL | 24(R2), -(SP) | | | | | | 0553 |
| | | 1A | A2 9F 00028 | | | PUSHAB | 26(R2) | | | | | | 0554 |
| | | | 52 DD 00028 | | | PUSHL | R2 | | | | | | 0555 |
| | | 0000V CF | 9F 0002D | | | PUSHAB | QIODEONE | | | | | | 0556 |
| | | 08 | A2 9F 00031 | | | PUSHAB | 8(R2) | | | | | | 0557 |
| | | | 30 DD 00034 | | | PUSHL | #48 | | | | | | 0558 |
| | 7E | 0000000G | 00 3C 00036 | | | MOVZWL | RDWRTCHAN, -(SP) | | | | | | 0559 |
| | | | 7E D4 0003D | | | CLRL | -(SP) | | | | | | 0560 |
| 0000000G | 00 | | 0C FB 0003F | | | CALLS | #12, SYSSQIO | | | | | | 0561 |
| | 63 | | 50 D0 00046 | | | MOVL | R0, RETSTATUS | | | | | | 0562 |
| | 1A | | 63 E8 00049 | | | BLBS | RETSTATUS, 2\$ | | | | | | 0563 |
| | | | 7E D4 0004C | | | CLRL | -(SP) | | | | | | 0564 |
| | 00 | | 01 FB 0004E | | | CALLS | #1, SYSSSETAST | | | | | | 0565 |
| | 0000000G | 00 | 01 90 00055 | | | MOVB | #1, WAKEFLAG | | | | | | 0566 |
| | | | 7E 7C 0005C | | | CLRQ | -(SP) | | | | | | 0567 |
| | 0000000G | 00 | 02 FB 0005E | | | CALLS | #2, SYSSWAKE | | | | | | 0568 |
| | | | 04 00065 | | | RET | | | | | | | 0569 |
| | 0000' CF | | 52 D0 00066 | 2\$: | | MOVL | R2, CURRENTIO | | | | | | 0570 |
| | | | 04 0006B | | | RET | | | | | | | 0571 |
| | 0000' DF | | 62 0E 0006C | 3\$: | | INSQUE | (R2), @IOQUEUE+4 | | | | | | 0572 |
| | | | 04 00071 | | | RET | | | | | | | 0573 |

; Routine Size: 114 bytes, Routine Base: \$CODE\$ + 029C

```
559      0553 1 ROUTINE READ(BUFFER): NOVALUE =
560      0554 1 ++
561      0555 1
562      0556 1 Functional Description:
563      0557 1 Perform a read QIO function to the terminal.
564      0558 1
565      0559 1 Calling Sequence:
566      0560 1 standard
567      0561 1
568      0562 1 Input Parameters:
569      0563 1 BUFFER = address of the link buffer
570      0564 1
571      0565 1 Implicit Inputs:
572      0566 1 CURRENTIO
573      0567 1 INDDATA
574      0568 1
575      0569 1 Output Parameters:
576      0570 1 none
577      0571 1
578      0572 1 Implicit Outputs:
579      0573 1 IOQUEUE
580      0574 1 CURRENTIO
581      0575 1 READINPROG
582      0576 1 UNSOLPEND
583      0577 1 Routines Called:
584      0578 1 INDREAD
585      0579 1 QIODONE
586      0580 1
587      0581 1 Routine Value:
588      0582 1 none
589      0583 1
590      0584 1 Signals:
591      0585 1 none
592      0586 1
593      0587 1 Side Effects:
594      0588 1 An I/O may be queued for later action.
595      0589 1
596      0590 1 --.
597      0591 2 BEGIN
598      0592 2 MAP BUFFER: REF RTP_BUF;
599      0593 2 LOCAL
600      0594 2     FUNCTION;
601      0595 2 IF .INDDATA NEQ 0 THEN
602      0596 3     BEGIN ! WE ALREADY HAVE INDIRECT COMMAND DATA
603      0597 3     BUFFER[RTP_IOS] = .INDDATA[RTP_IOS]; ! COPY THE IOSB
604      0598 3     BUFFER[RTP_IOC] = .INDDATA[RTP_IOC];
605      0599 3     CH$MOVE(.INDDATA[RTP_IOC]+1,INDDATA[RTP_DAT], ! COPY THE DATA
606      0600 3     BUFFER[RTP_DAT]);
607      0601 3     INDREAD(); ! LOOK FOR MORE DATA
608      0602 3     QIODONE(.BUFFER); ! PASS THIS DATA ON
609      0603 3     RETURN;
610      0604 2     END;
611      0605 2 IF .CURRENTIO EQL 0 THEN
612      0606 3     BEGIN
613      0607 3     IF (.BUFFER[RTP_MOD] AND RM_RBN) NEQ 0 THEN
614      0608 3         FUNCTION = IOS_TTYREADALL ! BINARY
615      0609 3     ELSE
```

```

616      0610 3
617      0611 3
618      P 0612 3
619      P 0613 3
620      P 0614 3
621      P 0615 3
622      P 0616 3
623      P 0617 3
624      P 0618 3
625      0619 3
626      0620 3
627      0621 3
628      0622 3
629      0623 3
630      0624 3
631      0625 2
632      0626 2
633      0627 1

        FUNCTION = IOS_READVBLK;      ! NORMAL
        RETSTATUS =
$Q10      (CHAN = .RDWRCHAN,      ! READ FROM THE TERMINAL
          FUNC = .FUNCTION+MAPMODIFIER(.BUFFER[RTP_MOD]),
          IOSB = BUFFER[RTP_IOS],
          ASTADR = QIODONE,
          ASTPRM = .BUFFER,
          P1 = BUFFER[RTP_DAT],
          P2 = .BUFFER[RTP_RCT],
          P4 = TERMINATOR(.BUFFER[RTP_MOD]));
QUIT ON ERROR;
CURRENTIO = .BUFFER;
UNSOLPEND = 0;      ! NO MORE DATA PENDING
READINPROG = 1;
END
ELSE
  INSQUE(.BUFFER,.IOQUEUE[1]);      ! QUEUE IT FOR LATER
END;

```

| | | | | | | | |
|----|----|--------------|------------------|------------------|---------------------------|----------------|------|
| | | | 01FC 00000 READ: | .WORD | Save R2,R3,R4,R5,R6,R7,R8 | : 0553 | |
| | | 58 00000000G | 00 9E 00002 | MOVAB | RETSTATUS, R8 | | |
| | | 57 0000 | CF 9E 00009 | MOVAB | CURRENTIO, R7 | | |
| | | 50 04 | A7 D0 0000E | MOVL | INDDATA, R0 | 0595 | |
| | | 56 04 | 22 13 00012 | BEQL | 1\$ | | |
| | 08 | A6 08 | AC D0 00014 | MOVL | BUFFER, R6 | 0597 | |
| | | 51 0A | A0 D0 00018 | MOVL | 8(R0), 8(R6) | | |
| | | | A0 3C 0001D | MOVZWL | 10(R0), R1 | 0599 | |
| | | | 51 D6 00021 | INCL | R1 | | |
| 1A | A6 | 1A A0 | 51 28 00023 | MOVC3 | R1, 26(R0), 26(R6) | 0600 | |
| | | 0000V CF | 00 FB 00029 | CALLS | #0, INDREAD | 0601 | |
| | | 0000V CF | 56 DD 0002E | PUSHL | R6 | 0602 | |
| | | | 01 FB 00030 | CALLS | #1, QIODONE | | |
| | | | 04 00035 | RET | | 0596 | |
| | | | 52 04 | AC D0 00036 1\$: | MOVL | BUFFER, R2 | 0607 |
| | | | 67 D5 0003A | TSTL | CURRENTIO | 0605 | |
| | | 05 | 11 A2 | 72 12 0003C | BNEQ | 5\$ | |
| | | | 53 02 | E1 0003E | BBC | #2 17(R2), 2\$ | 0607 |
| | | | 3A 00 | D0 00043 | MOVL | #58, FUNCTION | 0608 |
| | | | 03 11 | 00046 | BRB | 3\$ | |
| | | | 53 31 | D0 00048 2\$: | MOVL | #49, FUNCTION | 0610 |
| | | | 7E 7C | 00048 3\$: | CLRQ | -(SP) | 0619 |
| | | 0000V | CF 7E | 11 A2 9A 0004D | MOVZBL | 17(R2), -(SP) | |
| | | | | 01 FB 00051 | CALLS | #1, TERMINATOR | |
| | | | | 50 DD 00056 | PUSHL | R0 | |
| | | | | 7E D4 00058 | CLRL | -(SP) | |
| | | | 7E 16 | A2 3C 0005A | MOVZWL | 22(R2), -(SP) | |
| | | | 1A A2 | 9F 0005E | PUSHAB | 26(R2) | |
| | | | 52 DD | 00061 | PUSHL | R2 | |
| | | 0000V | CF 08 | 9F 00063 | PUSHAB | QIODONE | |
| | | | A2 11 | 9F 00067 | PUSHAB | 8(R2) | |
| | | 0000V | CF 7E | 9A 0006A | MOVZBL | 17(R2), -(SP) | |
| | | | | 01 FB 0006E | CALLS | #1 MAPMODIFIER | |
| | | | | 6043 9F 00073 | PUSHAB | (R0)[FUNCTION] | |

| | | | | |
|-----------|-------------|-------------|------------------------------|-------------------------|
| | 7E 0000000G | 00 | 3C 00076 | MOVZWL RDWRTCHAN, -(SP) |
| C0000000G | 00 | 7E D4 0007D | CLRL -(SP) | |
| | 68 | 0C FB 0007F | CALLS #12, SYSSQIO | |
| | 1A | 50 D0 00086 | MOVL R0 RETSTATUS | |
| 00000000G | 00 | 68 E8 00089 | BLBS RETSTATUS, 4\$ | |
| 00000000G | 00 | 7E D4 0008C | CLRL -(SP) | |
| 00000000G | 00 | 01 FB 0008E | CALLS #1, SYSSSETAST | |
| 00000000G | 00 | 01 90 00095 | MOVB #1 WAKEFLAG | |
| 00000000G | 00 | 7E 7C 0009C | CLRQ -(SP) | |
| | | 02 FB 0009E | CALLS #2, SYSSWAKE | |
| | | 04 000A5 | RET | |
| FA A7 | 0100 | 52 D0 000A6 | 4\$: MOVL R2 CURRENTIO | |
| | | 8F B0 000A9 | MOVW #256, UNSOLPEND | |
| | | 04 000AF | RET | |
| OC B7 | | 62 0E 000B0 | 5\$: INSQUE (R2), AIOQUEUE+4 | |
| | | 04 000B4 | RET | |

: Routine Size: 181 bytes. Routine Base: \$CODE\$ + 030E

```
: 635 0628 1 ROUTINE READPROMPT(BUFFER): NOVALUE =
: 636 0629 1 !++
: 637 0630 1
: 638 0631 1 Functional Description:
: 639 0632 1 Perform a readprompt QIO function to the terminal.
: 640 0633 1
: 641 0634 1 Calling Sequence:
: 642 0635 1 standard
: 643 0636 1
: 644 0637 1 Input Parameters:
: 645 0638 1 BUFFER = address of the link buffer
: 646 0639 1
: 647 0640 1 Implicit Inputs:
: 648 0641 1 CURRENTIO
: 649 0642 1 INDDATA
: 650 0643 1
: 651 0644 1 Output Parameters:
: 652 0645 1 none
: 653 0646 1
: 654 0647 1 Implicit Outputs:
: 655 0648 1 CURRENTIO
: 656 0649 1 UNSOLPEND
: 657 0650 1 READINPROG
: 658 0651 1 IOQUEUE
: 659 0652 1
: 660 0653 1 Routines Called:
: 661 0654 1 READ
: 662 0655 1
: 663 0656 1 Routine Value:
: 664 0657 1 none
: 665 0658 1
: 666 0659 1 Signals:
: 667 0660 1 none
: 668 0661 1
: 669 0662 1 Side Effects:
: 670 0663 1 An I/O may be queued for later action.
: 671 0664 1
: 672 0665 1 !--
: 673 0666 2 BEGIN
: 674 0667 2 MAP BUFFER: REF RTP_BUF;
: 675 0668 2 LOCAL
: 676 0669 2 FUNCTION;
: 677 0670 2 IF .INDDATA NEQ 0 THEN
: 678 0671 3 BEGIN ! WE HAVE INDIRECT COMMAND FILE DATA
: 679 0672 3 /READ T.BUFFER); ! GET THE DATA
: 680 0673 3 RETURN;
: 681 0674 2 END;
: 682 0675 2 IF .CURRENTIO EQL 0 THEN
: 683 0676 3 BEGIN
: 684 0677 3 IF (.BUFFER[RTP_MOD] AND RM_RBN) NEQ 0 THEN
: 685 0678 3 FUNCTION = IOS_TTYREADPALL ! BINARY
: 686 0679 3 ELSE
: 687 0680 3 FUNCTION = IOS_READPROMPT; ! NORMAL
: 688 0681 3 RETSTATUS =
: 689 P 0682 3 $QIO (CHAN = .RDWRCHAN, ! READPROMPT TO THE TERMINAL
: 690 P 0683 3 FUNC = .FUNCTION+MAPMODIFIER(.BUFFER[RTP_MOD]),
: 691 P 0684 3 IOSB = BUFFER[RTP_IOS].
```

```

: 692      P 0685 3
: 693      P 0686 3
: 694      P 0687 3
: 695      P 0688 3
: 696      P 0689 3
: 697      P 0690 3
: 698      P 0691 3
: 699      P 0692 3
: 700      P 0693 3
: 701      P 0694 3
: 702      P 0695 3
: 703      P 0696 3
: 704      P 0697 2
: 705      P 0698 2
: 706      P 0699 1

        ASTADR = QIODONE,
        ASTPRM = BUFFER,
        P1 = BUFFER[RTP_DAT],
        P2 = .BUFFER[RTP_RCT],
        P4 = TERMINATOR(.BUFFER[RTP_MOD]),
        P5 = BUFFER[RTP_DAT],
        P6 = .BUFFER[RTP_TCT]);

        QUIT ON ERROR;
        CURRENTIO = .BUFFER;
        UNSOLPEND = 0;           ! NO MORE DATA PENDING
        READINPROG = 1;
        END

    ELSE
        END;      INSQUE(.BUFFER,.IQUEUE[1]);   ! QUEUE IT FOR LATER

```

003C 00000 READPROMPT:

| | | | | | | |
|-----------|----|--------------|----------------|--------|------------------|------|
| | | | | .WORD | Save R2,R3,R4,R5 | 0628 |
| | | 55 00000000G | 00 9E 00002 | MOVAB | RETSTATUS, R5 | |
| | | 54 00000000' | CF 9E 00009 | MOVAB | CURRENTIO, R4 | |
| | | | 04 A4 D5 0000E | TSTL | INDDATA | 0670 |
| | | | 09 13 00011 | BEQL | 1\$ | |
| | | | 04 AC DD 00013 | PUSHL | BUFFER | 0672 |
| | | FF30 CF | 01 FB 00016 | CALLS | #1, READ | |
| | | | 04 04 0001B | RET | | 0671 |
| | | 52 | AC D0 0001C | MOVL | BUFFER, R2 | 0677 |
| | | | 64 D5 00020 | TSTL | CURRENTIO | 0675 |
| | | | 77 12 00022 | BNEQ | 5\$ | |
| 05 | 11 | A2 | 02 E1 00024 | BBC | #2, 17(R2), 2\$ | 0677 |
| | | 53 | 38 D0 00029 | MOVL | #59, FUNCTION | 0678 |
| | | | 03 11 0002C | BRB | 3\$ | |
| | | 53 | 37 D0 0002E | MOVL | #55, FUNCTION | 0680 |
| | | 7E | 18 A2 3C 00031 | MOVZWL | 24(R2), -(SP) | |
| | | | 1A A2 9F 00035 | PUSHAB | 26(R2) | 0691 |
| | | 0000V CF | 11 A2 9A 00038 | MOVZBL | 17(R2), -(SP) | |
| | | | 01 FB 0003C | CALLS | #1, TERMINATOR | |
| | | | 50 DD 00041 | PUSHL | R0 | |
| | | | 7E D4 00043 | CLRL | -(SP) | |
| | | 7E | 16 A2 3C 00045 | MOVZWL | 22(R2), -(SP) | |
| | | | 1A A2 9F 00049 | PUSHAB | 26(R2) | |
| | | | 52 DD 0004C | PUSHL | R2 | |
| | | 0000V CF | 9F 0004E | PUSHAB | QIODONE | |
| | | | 08 A2 9F 00052 | PUSHAB | 8(R2) | |
| | | 0000V CF | 11 A2 9A 00055 | MOVZBL | 17(R2), -(SP) | |
| | | | 01 FB 00059 | CALLS | #1, MAPMODIFIER | |
| | | | 6043 9F 0005E | PUSHAB | (R0)[FUNCTION] | |
| | | 7E 00000000G | 00 3C 00061 | MOVZWL | RDWRCHAN, -(SP) | |
| | | | 7E D4 00068 | CLRL | -(SP) | |
| 00000000G | 00 | | OC FB 0006A | CALLS | #12, SYSSQIO | |
| | 65 | | 50 D0 00071 | MOVL | R0 RETSTATUS | |
| | 1A | | 65 E8 00074 | BLBS | RESTATUS, 4\$ | |
| | | | 7E D4 00077 | CLRL | -(SP) | |
| | | 00000000G | 01 FB 00079 | CALLS | #1, SYSSSETAST | |

| | | |
|-------------|------------------------------|-------------------------|
| 0000000G 00 | 01 90 00080 | MOVB #1, WAKEFLAG |
| 0000000G 00 | 7E 7C 00087 | CLRR -(SP) |
| | 02 FB 00089 | CALLS #2, SY\$SWAKE |
| | 04 00090 | RET |
| FA A4 0100 | 52 D0 00091 4\$: 8F B0 00094 | MOVL R2, CURRENTIO |
| | 04 0009A | MOVW #256, UNSOLPEND |
| OC B4 | 62 0E 00098 5\$: 04 0009F | RET |
| | | INSQUE (R2), @IOQUEUE+4 |
| | | RET |

; Routine Size: 160 bytes, Routine Base: \$CODE\$ + 03C3

```
: 708 0700 1 ROUTINE QIODONE(BUFFER): NOVALUE =
: 709 0701 1 ++
: 710 0702 1
: 711 0703 1 Functional Description:
: 712 0704 1 Send a message on the link when a terminal QIO completes.
: 713 0705 1 Interpret the 'EXIT RMT' command to exit this program.
: 714 0706 1
: 715 0707 1 Calling Sequence:
: 716 0708 1 standard
: 717 0709 1
: 718 0710 1 Input Parameters:
: 719 0711 1 BUFFER = address of the link buffer.
: 720 0712 1
: 721 0713 1 Implicit Inputs:
: 722 0714 1 none
: 723 0715 1
: 724 0716 1 Output Parameters:
: 725 0717 1 none
: 726 0718 1
: 727 0719 1 Implicit Outputs:
: 728 0720 1 READINPROG
: 729 0721 1 CURRENTIO
: 730 0722 1 RETSTATUS
: 731 0723 1
: 732 0724 1 Routines Called:
: 733 0725 1 NEXTIO
: 734 0726 1 FREEBUF
: 735 0727 1
: 736 0728 1 Routine Value:
: 737 0729 1 none
: 738 0730 1
: 739 0731 1 Signals:
: 740 0732 1 none
: 741 0733 1
: 742 0734 1 Side Effects:
: 743 0735 1 If there is an error on the write to the link, a SWAKE will be issued
: 744 0736 1 to cause this program to abort.
: 745 0737 1
: 746 0738 1 --
: 747 0739 2 BEGIN
: 748 0740 2 MAP BUFFER: REF RTP_BUF;
: 749 0741 2 LOCAL
: 750 0742 2 COUNT: ;
: 751 0743 2 IF .BUFFER[RTP_IOS] AND 1 THEN
: 752 0744 2     BUFFER[RTP_STS] = RS_SFC                 ! GOOD STATUS
: 753 0745 2 ELSE
: 754 0746 2     BUFFER[RTP_STS] = RS_FPE;                 ! ERROR
: 755 0747 2 BUFFER[RTP_FLG] = 0;
: 756 0748 2 BUFFER[RTP_TCT] = 0;
: 757 0749 2 COUNT = 10;                                     ! MINIMUM MESSAGE LENGTH
: 758 0750 2 IF .BUFFER[RTP_FNC] NEQ RF_WTD THEN
: 759 0751 3     BEGIN                                     ! IT WAS A READ
: 760 0752 3         COUNT = .COUNT + .BUFFER[RTP_IOC]; ! ADD THE DATA
: 761 0753 3         IF (.BUFFER[RTP_MOD] AND (RM_RTC+RM_RNE)) EQL 0 THEN
: 762 0754 4             BEGIN                             ! CHECK FOR A CARRIAGE RETURN
: 763 0755 4                 IF (.BUFFER+.COUNT+16)<0,8> EQL 13 THEN
: 764 P 0756 4                 SQIOW (CHAN = .RDWRCHAN,                 ! ECHO CAR-RET
```

```

: 765 P 0757 4
: 766 P 0758 4
: 767 P 0759 4
: 768 0760 3
: 769 0761 3
: 770 0762 4
: 771 0763 4
: 772 0764 4
: 773 0765 4
: 774 0766 4
: 775 0767 3
: 776 0768 3
: 777 0769 3
: 778 0770 3
: 779 0771 3
: 780 0772 4
: 781 0773 4
: 782 0774 4
: 783 0775 3
: 784 C776 3
: 785 0777 3
: 786 0778 2
: 787 0779 2
: 788 0780 2
: 789 0781 2
: 790 0782 3
: 791 0783 3
: 792 0784 2
: 793 0785 2
: 794 0786 2
: 795 0787 3
: 796 0788 3
: P 0789 3
: P 0790 3
: P 0791 3
: P 0792 3
: P 0793 3
: P 0794 3
: P 0795 3
: P 0796 3
: P 0797 3
: P 0798 3
: P 0799 2
: 800 0799 2
: 801 0800 2
: 802 0801 1

        FUNC = IOS_WRITEVBLK,
        P1 = UPLIT(13),
        P2 = 1);

    END;
    IF .BUFFER[RTP_IOC] EQL 8 THEN ! COULD BE AN EXIT
        BEGIN
            IF CH$EQL(8,BUFFER[RTP_DAT],8,UPLIT('EXIT RMT')) THEN
                QUIT; ! GET OUT
            IF CH$EQL(8,BUFFER[RTP_DAT],8,UPLIT('exit rmt')) THEN
                QUIT; ! GET OUT
        END;
    IF .BUFFER[RTP_RCT] NEQ .BUFFER[RTP_IOC] THEN
        COUNT = .COUNT + 1 ! ADD TERMINATOR
    ELSE
        IF .COUNT NEQ 128 THEN
            BEGIN ! THIS IS A KLUGE FOR RSX
                COUNT = .COUNT+1;
                (.BUFFER+15+.COUNT)<0,8> = 0; ! ADD A NULL
            END;
        BUFFER[RTP_RCT] = .BUFFER[RTP_IOC]; ! COUNT
        READINPROG = 0; ! DONE
    END;
    IF ((.BUFFER[RTP_MOD] AND RM_WBT) EQL 0) AND
        (.BUFFER[RTP_FNC] NEQ RF_RSC) THEN
        CURRENTIO = 0; ! CURRENT I/O HAS COMPLETED
    IF ((.BUFFER[RTP_FNC] EQL RF_WTD) AND
        ((.BUFFER[RTP_MOD] AND RM_NWC) NEQ 0))
        OR (.BUFFER[RTP_IOS] EQL SSS_ABORT) THEN
        FREEBUF(.BUFFER)
    ELSE
        BEGIN
            RETSTATUS =
$QIO (CHAN = .LINKCHAN, ! WRITE TO ' NK
                FUNC = IOS_WRITEVBLK,
                IOSB = BUFFER[RTP_IOS],
                ASTADR = LINKWRDONE,
                ASTPRM = .BUFFER,
                P1 = BUFFER[RTP_FNC],
                P2 = .COUNT);
            IF .RETSTATUS EQL SSS_ABCRT THEN
                RETURN; ! Link gone - mailbox msg will tell why
            QUIT_ON_ERROR;
        END;
        NEXTIO(); ! CHECK FOR A PENDING I/O
    END;

```

.PSECT SPLITS,NOWRT,NOEXE,2

| | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|--------------|--------------|------------|----|
| 54 | 4D | 52 | 20 | 54 | 49 | 58 | 45 | 00000000 | 00030 P.AAD: | .LONG | 13 |
| 74 | 6D | 72 | 20 | 74 | 69 | 78 | 65 | 00034 P.AAE: | .ASCII | \EXIT RMT\ | |
| | | | | | | | | 0003C P.AAF: | .ASCII | \exit rmt\ | |

.PSECT SCODES,NOWRT,2

| | | | | | | | | |
|----------|-----------|-------|------|-------|------------------------|-------------------------|---------|--|
| | | | 007C | 00000 | Q1ODONE: .WORD | Save R2,R3,R4,R5,R6 | : 0700 | |
| | 54 | 04 | AC | D0 | 00002 | MOVL BUFFER, R4 | : 0743 | |
| | 56 | 10 | A4 | 9E | 00006 | MOVAB 16(R4), R6 | : 0744 | |
| | 05 | 08 | A4 | E9 | 0000A | BLBC 8(R4), 1\$ | : 0743 | |
| | | 03 | A6 | 94 | 0000E | CLRB 3(R6) | : 0744 | |
| | | | | 04 | 11 | 00011 | BRB 2\$ | |
| 03 | A6 | 01 | 90 | 00013 | 1\$: 2\$: | MOV B #1, 3(R6) | : 0746 | |
| | | 02 | A6 | 94 | 00017 | CLRB 2(R6) | : 0747 | |
| | | 18 | A4 | B4 | 0001A | CLRW 24(R4) | : 0748 | |
| | 55 | 0A | 00 | 0001D | | MOVL #10, COUNT | : 0749 | |
| | 03 | 66 | 91 | 00020 | | CMPB (R6), #3 | : 0750 | |
| | | 6E | 13 | 00023 | | BEQL 8\$ | | |
| | 50 | CA | A4 | 3C | 00025 | MOVZWL 10(R4), R0 | : 0752 | |
| | 55 | 50 | CO | 00029 | | ADDL2 R0, COUNT | | |
| | 18 | 01 | A6 | 93 | 0002C | BITB 1(R6), #24 | : 0753 | |
| | | 26 | 12 | 00030 | | BNEQ 3\$ | | |
| | 0D | 10 | A544 | 91 | 00032 | CMPB 16(COUNT)[R4], #13 | : 0755 | |
| | | | 1F | 12 | 00037 | BNEQ 3\$ | | |
| | | 7E | 7C | 00039 | | CLRQ -(SP) | : 0759 | |
| | | 7E | 7C | 0003B | | CLRQ -(SP) | | |
| | | 01 | DD | 0003D | | PUSHL #1 | | |
| | | CF | 9F | 0003F | | PUSHAB P.AAD | | |
| | | 7E | 7C | 00043 | | CLRQ -(SP) | | |
| | | 7E | 30 | 7D | 00045 | MOVQ #48, -(SP) | | |
| | | 7E | 00 | 3C | 00048 | MOVZWL RDWRCHAN, -(SP) | | |
| | 00000000G | 00 | 0C | F8 | 00051 | CLRL -(SP) | | |
| | | 08 | A4 | B1 | 00058 | CALLS #12, SYSSQIOW | | |
| | | | 12 | 12 | 0005C | CMPW 10(R4), #8 | : 0761 | |
| 0000' CF | 1A | A4 | 08 | 29 | 0005E | BNEQ 5\$ | | |
| 0000' CF | 1A | A4 | 07 | 13 | 00065 | CMPC3 #8, 26(R4), P.AAE | : 0763 | |
| | | 08 | 29 | 00067 | | BEQL 4\$ | | |
| | 0A | A4 | 7A | 13 | 0006E | CMPC3 #8, 26(R4), P.AAF | : 0765 | |
| | | 16 | A4 | 81 | 00070 | 4\$: BEQL 13\$ | | |
| | | | 04 | 13 | 00075 | CMPW 22(R4), 10(R4) | : 0768 | |
| | | | 55 | 06 | 00077 | BEQL 6\$ | | |
| | | | 0F | 11 | 00079 | INCL COUNT | : 0769 | |
| | 00000080 | 8F | 55 | D1 | 0007B | BRB 7\$ | | |
| | | | 06 | 13 | 00082 | CMPL COUNT, #128 | : 0771 | |
| | | | 55 | D6 | 00084 | BEQL 7\$ | | |
| | | | 0F | A544 | 94 | INCL COUNT | : 0773 | |
| | | 16 | A4 | 80 | 00086 | CLRB 15(COUNT)[R4] | : 0774 | |
| | | 0A | A4 | 94 | 0008A | MOVW 10(R4), 22(R4) | : 0776 | |
| | 09 | 0000' | CF | 94 | 0008F | CLRB READINPROG | : 0777 | |
| | 66 | 09 | E0 | 00093 | 8\$: BBS #9, (R6), 9\$ | : 0779 | | |
| | 07 | 66 | 91 | 00097 | | CMPB (R6), #7 | : 0780 | |
| | | 04 | 13 | 0009A | BEQL 9\$ | | | |
| | 03 | 0000' | CF | D4 | 0009C | CLRL CURRENTIO | : 0781 | |
| | | 66 | 91 | 000A0 | 9\$: CMPB (R6), #3 | : 0782 | | |
| | | 04 | 12 | 000A3 | BNEQ 10\$ | | | |
| | | 66 | B5 | 000A5 | TSTW (R6) | : 0783 | | |
| | | 06 | 19 | 000A7 | BLSS 11\$ | | | |
| | 2C | 08 | A4 | B1 | 000A9 | 10\$: CMPW 8(R4), #44 | : 0784 | |
| | | | 09 | 12 | 000AD | BNEQ 12\$ | | |
| | | 54 | DD | 000AF | 11\$: PUSHL R4 | : 0785 | | |
| | FC94 | CF | 01 | FB | 000B1 | CALLS #1, FREEBUF | | |
| | | | 4C | 11 | 000B6 | BRB 14\$ | | |

| | | | | |
|--|--------------|-------------------|------------------------|--------|
| | | 7E 7C 000B8 12\$: | CLRQ -(SP) | : 0795 |
| | | 7E 7C 000BA | CLRQ -(SP) | |
| | | 55 DD 000BC | PUSHL COUNT | |
| | 0050 | 8F BB 000BE | PUSHR #^M<R4,R6> | |
| | 0000V | CF 9F 000C2 | PUSHAB LINKWR1DONE | |
| | 08 | A4 9F 000C6 | PUSHAB 8(R4) | |
| | | 30 DD 000C9 | PUSHL #48 | |
| | | 7E 00000000G 00 | MOVZWL LINKCHAN, -(SP) | |
| | | 00 3C 000CB | CLRL -(SP) | |
| | | 7E D4 000D2 | CALLS #12, SYSSQIO | |
| | 00000000G 00 | 0C FB 000D4 | MOVL R0, RETSTATUS | |
| | | 50 D0 000DB | CMPL R0, #44 | |
| | | 2C 50 D1 000E2 | BEQL 15\$ | 0796 |
| | | 1A 22 13 000E5 | BLBS R0, 14\$ | 0797 |
| | 00000000G 00 | 50 E8 000E7 | CLRL -(SP) | |
| | | 7E D4 000EA 13\$: | CALLS #1, SYSSSETAST | |
| | 00000000G 00 | 01 FB 000EC | MOVB #1, WAKEFLAG | |
| | | 01 90 000F3 | CLRQ -(SP) | |
| | 00000000G 00 | 7E 7C 000FA | CALLS #2, SYSSWAKE | |
| | | 02 FB 000FC | RET | |
| | | 04 00103 | CALLS #0, NEXTIO | |
| | 0000V CF | 00 FB 00104 14\$: | RET | 0800 |
| | | 04 00109 15\$: | | 0801 |

: Routine Size: 266 bytes, Routine Base: \$CCDES + 0463

: 810 0802 1

```

: 812      0803 1 ROUTINE LINKWRTDONE(BUFFER): NOVALUE =
: 813      0804 1 ++
: 814      0805 1
: 815      0806 1 Functional Description:
: 816      0807 1     Free the link buffer when a write to the link completes
: 817      0808 1
: 818      0809 1 Calling Sequence:
: 819      0810 1     standard
: 820      0811 1
: 821      0812 1 Input Parameters:
: 822      0813 1     BUFFER = address of the link buffer.
: 823      0814 1
: 824      0815 1 Implicit Inputs:
: 825      0816 1     none
: 826      0817 1
: 827      0818 1 Output Parameters:
: 828      0819 1     RETSTATUS
: 829      0820 1
: 830      0821 1 Implicit Outputs:
: 831      0822 1     none
: 832      0823 1
: 833      0824 1 Routines Called:
: 834      0825 1     FREEBUF
: 835      0826 1
: 836      0827 1 Routine Value:
: 837      0828 1     none
: 838      0829 1
: 839      0830 1 Signals.
: 840      0831 1     none
: 841      0832 1
: 842      0833 1 Side Effects:
: 843      0834 1     If there was an error on the write to the link, a SWAKE is issued to
: 844      0835 1     cause the program to abort.
: 845      0836 1
: 846      0837 1 --+
: 847      0838 2 BEGIN
: 848      0839 2     MAP BUFFER: REF RTP BUF;
: 849      0840 2     RETSTATUS = .BUFFER[RTP IOS];
: 850      0841 2     IF .RETSTATUS EQL SSS_ABORT THEN
: 851      0842 2       RETURN;           ! Link gone - mailbox msg will tell why
: 852      0843 2     QUIT ON ERROR;
: 853      0844 2     FREEBUF(.BUFFER);    ! WE NO LONGER NEED THE BUFFER
: 854      0845 1     END;

```

000C 00000 LINKWRTDONE:

| | | | | | | | |
|----|-----------|----|-------|--------|------------------|---------------|--------|
| 53 | 00000000G | 00 | 9E | 00002 | .WORD | Save R2,R3 | : 0803 |
| 52 | 04 | AC | D0 | 00009 | MOVAB | RETSTATUS, R3 | : 0840 |
| 63 | 08 | A2 | 3C | 0000D | MOVL | BUFFER, R2 | : 0841 |
| 50 | 63 | D0 | 00011 | MOVZWL | 8(R2), RETSTATUS | | |
| 2C | 50 | D1 | 00014 | MOVL | RETSTATUS, R0 | | |
| | 24 | 13 | 00017 | CMPL | R0, #44 | | |
| 1A | 50 | E8 | 00019 | BEQL | 2\$ | | |
| | | | | BLBS | R0, 1\$ | : 0842 | |

| | | |
|--------------|------------------------------|----------------------|
| 00000000G 00 | 7E D4 0001C | CLRL -(SP) |
| 00000000G 00 | 01 FB 0001E | CALLS #1, SYSSSETAST |
| 00000000G 00 | 01 90 00025 | MOVB #1, WAKEFLAG |
| | 7E 7C 0002C | CLRQ -(SP) |
| | 02 FB 0002E | CALLS #2, SYSSWAKE |
| | 04 00035 | RET |
| FC03 CF | 52 DD 00036 1\$: 01 FB 00038 | PUSHL R2 |
| | 04 0003D 2\$: | CALLS #1, FREEBUF |
| | | RET |

0844
0845

: Routine Size: 62 bytes. Routine Base: \$CODE\$ + 056D

```
: 856    0846 1 ROUTINE UNSDatenBL(BUFFER): NOVALUE =
: 857    0847 1 ++
: 858    0848 1
: 859    0849 1 Functional Description:
: 860    0850 1   Enable or disable unsolicited data to the RSX system.
: 861    0851 1
: 862    0852 1 Calling Sequence:
: 863    0853 1   standard
: 864    0854 1
: 865    0855 1 Input Parameters:
: 866    0856 1   BUFFER = address of the link buffer
: 867    0857 1
: 868    0858 1 Implicit Inputs:
: 869    0859 1   UNSOLPEND
: 870    0860 1   INDDATA
: 871    0861 1
: 872    0862 1 Output Parameters:
: 873    0863 1   none
: 874    0864 1
: 875    0865 1 Implicit Outputs:
: 876    0866 1   UNSOLENBLFLG
: 877    0867 1
: 878    0868 1 Routines Called:
: 879    0869 1   TERMMBXMSG
: 880    0870 1   FREEBUF
: 881    0871 1   READ
: 882    0872 1
: 883    0873 1 Routine Value:
: 884    0874 1   none
: 885    0875 1
: 886    0876 1 Signals:
: 887    0877 1   none
: 888    0878 1
: 889    0879 1 Side Effects:
: 890    0880 1   If unsolicited input is enabled, any pending data is read.
: 891    0881 1
: 892    0882 1 --
: 893    0883 2 BEGIN
: 894    0884 2   MAP BUFFER: REF RTP_BUF;
: 895    0885 2   LOCAL
: 896    0886 2     NEWBUF: REF VECTOR;
: 897    0887 2     IF .BUFFER[RTP_FLG] NEQ RM_TUI THEN
: 898    0888 3       BEGIN
: 899    0889 3         IF .INDDATA NEQ 0 THEN
: 900    0890 4           BEGIN
: 901    0891 4             NEWBUF = GETBUF(); ! THERE IS INDIRECT FILE DATA
: 902    0892 4             CH$MOVE(40,.BUFFER,.NEWBUF); ! GET A SUBSTITUTE BUFFER
: 903    0893 4             READ (.NEWBUF); ! COPY HEADER + SOME
: 904    0894 3             END;
: 905    0895 3         UNSOLENBLFLG = .BUFFER; ! GET IT
: 906    0896 3         IF .UNSOLPEND NEQ 0 THEN ! ENABLE
: 907    0897 3           TERMMBXMSG(); ! DATA ALREADY PENDING
: 908    0898 3
: 909    0899 2       END
: 910    0900 3       BEGIN
: 911    0901 3         FREEBUF(.BUFFER); ! DISABLE
: 912    0902 3         IF .UNSOLEBLFLG NEQ 0 THEN ! NO LONGER NEED BUFFER
```

```
:
913 0903 4
914 0904 4
915 0905 4
916 0906 3
917 0907 2
918 0908 1
```

```
BEGIN
FREEBUF(.UNSOLENBLFLG); ' UNSOL DATA BUFFER
UNSOL+NBLFLG = 0;
END;
END;
```

| 01FC 00000 UNSDATEENBL: | | | | | | |
|-------------------------|-------|-------|----------------|------------|---------------------------|--------|
| | | | | .WORD | Save R2,R3,R4,R5,R6,R7,R8 | : 0846 |
| | 58 | 0000' | CF 9E 00002 | MOVAB | UNSOLENBLFLG, R8 | : 0887 |
| | 56 | 04 | AC D0 00007 | MOVL | BUFFER, R6 | : 0889 |
| | 80 | 8F | 12 A6 91 0000B | CMPB | 18(R6), #128 | : 0891 |
| | | | 26 13 00010 | BEQL | 2\$ | : 0892 |
| | | | 10 A8 05 00012 | TSTL | INDDATA | : 0893 |
| | | | 13 13 00015 | BEQL | 1\$ | : 0895 |
| | FBC2 | CF | 00 FB 00017 | CALLS | #0, GETBUF | : 0896 |
| | 57 | | 50 D0 0001C | MOVL | R0, NEWBUF | : 0897 |
| 67 | 66 | | 28 28 0001F | MOVC3 | #40, (R6), (NEWBUF) | : 0898 |
| | | | 57 DD 00023 | PUSHL | NEWBUF | : 0899 |
| | FD39 | CF | 01 FB 00025 | CALLS | #1, READ | : 0901 |
| | 68 | | 56 D0 0002A | 1\$: MOVL | R6, UNSOLENBLFLG | : 0902 |
| | | | 18 95 0002D | TSTB | UNSOLENPEND | : 0904 |
| | | 06 | 1B 13 00030 | BEQL | 3\$ | : 0905 |
| | 0000V | CF | 00 FB 00032 | CALLS | #2, TERMMBXMSG | : 0906 |
| | | | 04 00037 | RET | | : 0908 |
| | FBC3 | CF | 56 DD 00038 | 2\$: PUSHL | Fc | : 0909 |
| | 50 | | 01 FB 0003A | CALLS | #1, FREEBUF | : 0910 |
| | | | 68 D0 0003F | MOVL | UNSOLENBLFLG, R0 | : 0911 |
| | | | 09 13 00042 | BEQL | 3\$ | : 0912 |
| | FBB7 | CF | 50 DD 00044 | PUSHL | R0 | : 0913 |
| | | | 01 FB 00046 | CALLS | #1, FREEBUF | : 0914 |
| | | | 68 D4 0004B | CLRL | UNSOLENBLFLG | : 0915 |
| | | | 04 0004D | 3\$: RET | | : 0916 |

: Routine Size: 78 bytes. Routine Base: \$CODE\$ + 05AB

```
0909 1 ROUTINE TERMMBXMSG: NOVALUE =
0910 1 ++
0911 1 Functional Description:
0912 1 Handle messages from the terminal mailbox indicating unsolicited data
0913 1 or hangup.
0914 1
0915 1 Calling Sequence:
0916 1 standard
0917 1
0918 1 Input Parameters:
0919 1 none
0920 1
0921 1 Implicit Inputs:
0922 1 READINPROG
0923 1 UNSOLENBLFLG
0924 1 ATTACHFLAG
0925 1 SINGLEFLAG
0926 1
0927 1
0928 1 Output Parameters:
0929 1 none
0930 1
0931 1 Implicit Outputs:
0932 1 UNSOLENBLFLG
0933 1 SINGLEINPROG
0934 1 UNSOLPEND
0935 1
0936 1 Routines Called:
0937 1 GETBUF
0938 1
0939 1 Routine Value:
0940 1 none
0941 1
0942 1 Signals:
0943 1 none
0944 1
0945 1 Side Effects:
0946 1 In the case of unsolicited input, a read to the terminal is initiated
0947 1 if either unsolicited input or single character mode is enabled. A
0948 1 new read to the terminal mailbox is also initiated.
0949 1 In the case of a hangup, a SWAKE is issued to cause the program to
0950 1 abort.
0951 1
0952 1 --
0953 2 BEGIN
0954 2 MAP UNSOLENBLFLG: REF VECTOR;
0955 2 LOCAL
0956 2 NEWBUF: REF VECTOR;
0957 2 IF .TERMMBXDATA[0] EQ MSG$_TRMUNSOLIC THEN
0958 3 BEGIN
0959 3 IF .READINPROG EQ 0 THEN
0960 4 BEGIN
0961 4 IF (.UNSOLEBLFLG NEQ 0) AND
0962 4 ((.ATTACHFLAG OR .SINGLEFLAG) EQ 0) THEN
0963 5 BEGIN
0964 5 READ(.UNSOLENBLFLG); ! READ IT
0965 5 NEWBUF = GETBUF(); ! GET ANOTHER BUFFER
```

```

977      0966 5          NEWBUF[4] = .UNSOLENBLFLG[4];
978      0967 5          NEWBUF[5] = .UNSOLENBLFLG[5];
979      0968 5          UNSOLENBLFLG = .NEWBUF;
980      0969 5          END
981      0970 4          ELSE IF .SINGLEFLAG NEQ 0 THEN
982      0971 5          BEGIN ! READ A SINGLE CHARACTER
983      0972 5          RETSTATUS =
984      P 0973 5          SQIO (CHAN = RDWRCHAN,
985      P 0974 5          FUNC = IOS READVBLK+IOSM_BINARY+
986      P 0975 5          MAPMODIFIER(.SINGEFLAG[RTP_MOD]),
987      P 0976 5          IOSB = SINGLEFLAG[RTP_IOS],
988      P 0977 5          ASTADR = ONECHAR,
989      P 0978 5          ASTPRM = .SINGLEFLAG,
990      P C979 5          P1 = SINGLEFLAG[RTP_DAT],
991      0980 5          P2 = 1);
992      0981 5          QUIT_ON_ERROR;
993      0982 5          SINGEFLAGPROG = 1;
994      0983 5          UNSOLPEND = 0; ! NO MORE DATA PENDING
995      0984 5          END
996      0985 4          ELSE
997      0986 4          UNSOLPEND = 1; ! UNSOLICITED DATA PENDING
998      0987 3          END;
999      0988 3          RETSTATUS =
1000     P 0989 3          SQIO (CHAN = TERMMBXCHAN, ! DO IT AGAIN
1001     P 0990 3          FUNC = IOS READVBLK;
1002     P 0991 3          ASTADR = TERMMBXMSG;
1003     P 0992 3          P1 = TERMMBXDATA,
1004     0993 3          P2 = 8);
1005     0994 3          QUIT_ON_ERROR;
1006     0995 3          END
1007     0996 2          ELSE
1008     0997 3          QUIT ! HANGUP - SO QUIT
1009     0998 1          END:

```

001C 00000 TERMMBXMSG:

| | | | | |
|--------------|----------------|----------------|------------------|------|
| 54 00000000G | 00 9E 00002 | .WORD | Save R2,R3,R4 | 0909 |
| 53 00000000G | 00 9E 00009 | MOVAB | SYSSQIO, R4 | |
| 52 0000' F8 | A2 B1 00010 | MOVAB | RETSTATUS, R3 | |
| 01 | 03 13 00019 | CMPW | UNSOLENBLFLG, R2 | |
| | 0091 31 0001B | BEQL | TERMMBXDATA, #1 | |
| | 07 A2 95 0001E | BRW | 1\$ | |
| | 68 12 00021 | BNEQ | 5\$ | |
| 51 | 62 D0 00023 | MOVL | READINPROG | 0959 |
| | 23 13 00026 | BEQL | 4\$ | |
| 50 | 04 A2 9A 00028 | MOVZBL | UNSOLENBLFLG, R1 | 0961 |
| 50 | 08 A2 C8 0002C | R'2 | ATTACHFLAG, R0 | 0962 |
| | 19 12 00030 | SINGLEFLAG, R0 | | |
| | 51 DD 00032 | + | 2\$ | |
| FCDC | CF 01 FB 00034 | FUSHL | R1 | 0964 |
| FB52 | CF 00 FB 00039 | CALLS | #1, READ | |
| | 51 62 D0 0003E | CALLS | #0, GETBUF | 0965 |
| | | MOVL | UNSOLENBLFLG, R1 | 0966 |

| | | | | | | | | | | | |
|----|----|----|-------|-----------|-----------|-----------|--------|----------------------|------------------|----------------|--------------------|
| 10 | A0 | 10 | A1 | 7D | 00041 | | MOVQ | 16(R1), 16(NEWBUF) | | | |
| | 62 | | 50 | D0 | 00046 | | MOVL | NEWBUF, UNSOLENBLFLG | 0968 | | |
| | | | 40 | 11 | 00049 | | BRB | 4\$ | 0961 | | |
| | 50 | 08 | A2 | D0 | 0004B | 2\$: | MOVL | SINGLEFLAG, R0 | 0970 | | |
| | | | 36 | 13 | 0004F | | BEQL | 3\$ | | | |
| | | | 7E | 7C | 00051 | | CLRQ | -(SP) | | | |
| | | | 7E | 7C | 00053 | | CLRQ | -(SP) | 0980 | | |
| | | | 01 | DD | 00055 | | PUSHL | #1 | | | |
| | | 1A | A0 | 9F | 00057 | | PUSHAB | 26(R0) | | | |
| | | | 50 | DD | 0005A | | PUSHL | R0 | | | |
| | | | 0000V | CF | 9F | 0005C | PUSHAB | ONECHAR | | | |
| | | | 08 | A0 | 9F | 00060 | PUSHAB | 8(R0) | | | |
| | | | 7E | 11 | A0 | 9A | 00063 | MOVZBL | 17(R0), -(SP) | | |
| | | | CF | | 01 | FB | 00067 | CALLS | #1, MAFMODIFIER | | |
| | | | | 71 | A0 | 9F | 0006C | PUSHAB | 113(R0) | | |
| | | | 7E | 00000000G | 00 | 3C | 0006F | MOVZWL | RDWRCHAN, -(SP) | | |
| | | | | | 7E | D4 | 00076 | CLRL | -(SP) | | |
| | | | 64 | | 0C | FB | 00078 | CALLS | #12, SYSSQIO | | |
| | | | 63 | | 50 | DD | 0007B | MOVL | R0, RETSTATUS | | |
| | | | 2E | | 63 | E9 | 0007E | BLBC | RESTATUS, 5\$ | | |
| | | 05 | A2 | | 01 | B0 | 00081 | MOVW | #1, SINGLEINPROG | | |
| | | | | | 04 | 11 | 00085 | BRB | 4\$ | | |
| | | 06 | A2 | | 01 | 90 | 00087 | 5\$: | MOVB | #1, UNSOLPEND | |
| | | | | | 7E | 7C | 0008B | 4\$: | CLRQ | -(SP) | |
| | | | | | 7E | 7C | 0008D | CLRQ | -(SP) | | |
| | | | | | 08 | DD | 0008F | PUSHL | #8 | | |
| | | | | F8 | A2 | 9F | 00091 | PUSHAB | TERMMBXDATA | | |
| | | | | | 7E | D4 | 00094 | CLRL | -(SP) | | |
| | | | | FF66 | CF | 9F | 00096 | PUSHAB | TERMMBXMSG | | |
| | | | | 7E | 00000000G | 31 | 7D | 0009A | MOVQ | #49, -(SP) | |
| | | | | | 7E | 00000000G | 00 | 3C | 0009D | MOVZWL | TERMMBXCHAN, -(SP) |
| | | | | | | 7E | D4 | 000A4 | CLRL | -(SP) | |
| | | | 64 | | 0C | FB | 000A6 | CALLS | #12, SYSSQIO | | |
| | | | 63 | | 50 | DD | 000A9 | MOVL | R0, RETSTATUS | | |
| | | | 19 | | 63 | E8 | 000AC | BLBS | RESTATUS, 6\$ | | |
| | | | | 00000000G | 00 | 7E | D4 | 000AF | 5\$: | CLRL | -(SP) |
| | | | | 00000000G | 00 | 01 | FB | 000B1 | CALLS | #1, SYSSSETAST | |
| | | | | 00000000G | 00 | 01 | 90 | 000B8 | MOVB | #1, WAKEFLAG | |
| | | | | | | 7E | 7C | 000BF | CLRQ | -(SP) | |
| | | | | | | 02 | FB | 000C1 | CALLS | #2, SYSSWAKE | |
| | | | | | | 04 | 000C8 | 6\$: | RET | | |
| | | | | | | | | | | 0996 | |
| | | | | | | | | | | : | |
| | | | | | | | | | | 0998 | |
| | | | | | | | | | | : | |

: Routine Size: 201 bytes, Routine Base: \$CODE\$ + 05F9

```

: 1011      0999 1 ROUTINE BROADCAST(BUFFER): NOVALUE =
: 1012      1000 1 ++
: 1013      1001 1
: 1014      1002 1 Functional Description:
: 1015      1003 1 Issue a broadcast function to the terminal.
: 1016      1004 1
: 1017      1005 1 Calling Sequence:
: 1018      1006 1 standard
: 1019      1007 1
: 1020      1008 1 Input Parameters:
: 1021      1009 1 BUFFER = address of the link buffer
: 1022      1010 1
: 1023      1011 1 Implicit Inputs:
: 1024      1012 1 none
: 1025      1013 1
: 1026      1014 1 Output Parameters:
: 1027      1015 1 none
: 1028      1016 1
: 1029      1017 1 Implicit Outputs:
: 1030      1018 1 none
: 1031      1019 1
: 1032      1020 1 Routines Called:
: 1033      1021 1 QIODONE
: 1034      1022 1
: 1035      1023 1 Routine Value:
: 1036      1024 1 none
: 1037      1025 1
: 1038      1026 1 Signals:
: 1039      1027 1 none
: 1040      1028 1
: 1041      1029 1 Side Effects:
: 1042      1030 1 none
: 1043      1031 1
: 1044      1032 1 --
: 1045      1033 2 BEGIN
: 1046      1034 2 MAP BUFFER: REF RTP_BUF;
: 1047      1035 2 LOCAL
: 1048      1036 2     BRDCSTDESC: VECTOR[2];
: 1049      1037 2     BRDCSTDESC[0] = .BUFFER[RTP TCT];           ! COUNT
: 1050      1038 2     BRDCSTDESC[1] = BUFFER[RTP DAT];           ! DATA ADDRESS
: 1051      P 1039 2     BUFFER[RTP_IOS] = $BRDCST ?MSGBUF = BRDCSTDESC, ! BROADCAST IT
: 1052      1040 2             DEVNAM = TTYDESC);
: 1053      1041 2     QIODONE(.BUFFER);                      ! CLEAN UP
: 1054      1042 1     END;

```

.EXTRN SYS\$BRDCST

0004 00000 BROADCAST:

| | | | |
|----------------------------------|--|--|------------------------------|
| 04 5E 04 52 04 7E 04 AE | 04 04 C2 00002 04 AC D0 00005 18 A2 3C 00009 1A A2 9E 0000D 20 DD 00012 7E D4 00014 | .WORD Save R2 SUPL2 #4, SP MCVL BUFFER, R2 MOVZWL 24(R2), BRDCSTDESC MOVAB 26(R2), BRDCSTDESC+4 PUSHL #32 CLRL -(SP) | 0999 1037 1038 1040 |
|----------------------------------|--|--|------------------------------|

D 5
16-Sep-1984 02:18:51
14-Sep-1984 13:04:57VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RTPAD.SRC]RSXRT.B32;1Page 38
(13)

| | | | | | |
|-----------|----|----|----------|--------|----------------|
| 00000000G | 00 | 9F | 00016 | PUSHAB | TTYDESC |
| 00000000G | 00 | AE | 0001C | PUSHAB | BRDCSTDESC |
| 08 | A2 | 04 | FB 0001F | CALLS | #4, SYSSBRDCST |
| | | 50 | B0 00026 | MOVW | R0, 8(R2) |
| | | 52 | DD 0002A | PUSHL | R2 |
| FD70 | CF | 01 | FB 0002C | CALLS | #1, QIODONE |
| | | 04 | 00031 | RET | |

; 1041
; 1042

: Routine Size: 50 bytes, Routine Base: \$CODE\$ + 06C2

: 1055 1043 1

: 1057 1044 1 ROUTINE CNTRLCAST: NOVALUE =
: 1058 1045 1 ++
: 1059 1046 1 Functional Description:
: 1060 1047 1 Handle the AST indicating that a control-C was typed on the terminal.
: 1061 1048 1 Calling Sequence:
: 1062 1049 1 standard
: 1063 1050 1 Input Parameters:
: 1064 1051 1 none
: 1065 1052 1 Implicit Inputs:
: 1066 1053 1 none
: 1067 1054 1 Output Parameters:
: 1068 1055 1 none
: 1069 1056 1 Implicit Outputs:
: 1070 1057 1 none
: 1071 1058 1 Routines Called:
: 1072 1059 1 none
: 1073 1060 1 Routine Value:
: 1074 1061 1 none
: 1075 1062 1 Signals:
: 1076 1063 1 none
: 1077 1064 1 Side Effects:
: 1078 1065 1 A message is sent to the host and the control-C AST is enabled. An
: 1079 1066 1 error will cause a \$WAKE to be issued to abort the program.
: 1080 1067 1
: 1081 1068 1
: 1082 1069 1
: 1083 1070 1
: 1084 1071 1
: 1085 1072 1
: 1086 1073 1
: 1087 1074 1
: 1088 1075 1 --
: 1089 1076 1 BEGIN
: 1090 1077 1 RETSTATUS =
: 1091 1078 1 P 1080 2 SQIO (CHAN = .LINKCHAN, ! TELL HOST
: 1092 1079 1
: 1093 1081 2 FUNC = IOS_WRITEVBLK,
: 1094 1082 2 P1 = CNTRLMSG,
: 1095 1083 2 P2 = 4);
: 1096 1084 2 QUIT_ON_ERROR;
: 1097 1085 2 RETSTATUS =
: 1098 1086 2 P 1087 2 SQIO (CHAN = .CNTRLCHAN, ! REENABLE IT
: 1099 1087 2 FUNC = IOS_SETMODE+IOSM_CTRLCAST,
: 1100 1088 2 P1 = CNTRLCAST);
: 1101 1089 2 QUIT_ON_ERROR;
: 1102 1090 2 END;
: 1103 1091 1

000C 00000 CNTRLCAST:

| | | | | |
|--------------|----|----------|-------|---------------|
| 53 00000000G | 00 | 9E 00002 | .WORD | Save R2,R3 |
| 52 00000000G | 00 | 9E 00009 | MOVAB | SYSSQIO, R3 |
| | | | MOVAB | RETSTATUS, R2 |

: 1044

F 5
16-Sep-1984 02:18:51
14-Sep-1984 13:04:57VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RTPAD.SRC]RSXRT.B32;1Page 40
(14)

| | | | | | | |
|--------------|----|-------|----------|--------|-------------------|------|
| | 7E | 7C | 00010 | CLRQ | -(SP) | 1084 |
| | 7E | 7C | 00012 | CLRQ | -(SP) | |
| | 04 | DD | 00014 | PUSHL | #4 | |
| | CF | 9F | 00016 | PUSHAB | CNTRL(CMSG | |
| 0000' | 7E | 7C | 0001A | CLRQ | -(SP) | |
| | 30 | 7D | 0001C | MOVQ | #48, -(SP) | |
| | 7E | D4 | 00026 | MOVZWL | LINK(CHAN, -(SP) | |
| 7E 00000000G | 00 | 3C | 0001F | CLRL | -(SP) | |
| | OC | FB | 00028 | CALLS | #12, SYSSQIO | |
| 63 | 50 | DO | 0002B | MOVL | R0, RETSTATUS | |
| 62 | 62 | E9 | 0002E | BLBC | RESTATUS, 1\$ | |
| 24 | 7E | 7C | 00031 | CLRQ | -(SP) | 1089 |
| | 7E | 7C | 00033 | CLRQ | -(SP) | |
| | 7E | D4 | 00035 | CLRL | -(SP) | |
| | C6 | AF | 00037 | PUSHAB | CNTRL(CAST | |
| | 7E | 7C | 0003A | CLRQ | -(SP) | |
| | 7E | D4 | 0003C | CLRL | -(SP) | |
| 7E 0123 | 8F | 3C | 0003E | MOVZWL | #291, -(SP) | |
| 7E 00000000G | 00 | 3C | 00043 | MOVZWL | CNTRL(CHAN, -(SP) | |
| | 7E | D4 | 0004A | CLRL | -(SP) | |
| 63 | OC | FB | 0004C | CALLS | #12, SYSSQIO | |
| 62 | 50 | DO | 0004F | MOVL | R0, RETSTATUS | |
| 19 | 62 | E8 | 00052 | BLBS | RESTATUS, 2\$ | |
| 00000000G | 00 | 7E | 00055 | CLRL | -(SP) | |
| 00000000G | 00 | 01 | FB 00057 | CALLS | #1, SYSSSETAST | |
| 00000000G | 00 | 01 | 90 0005E | MOVB | #1, WAKEFLAG | |
| | 7E | 7C | 00065 | CLRQ | -(SP) | |
| | 02 | FB | 00067 | CALLS | #2, SYSSWAKE | |
| | 04 | 0006E | 2\$: | RET | | 1091 |

: Routine Size: 111 bytes, Routine Base: \$CODE\$ + 06F4

```

: 1106    1092 1 ROUTINE CNTRLYAST: NOVALUE =
: 1107    1093 1 ++
: 1108    1094 1
: 1109    1095 1 Functional Description:
: 1110    1096 1 Handle the AST indicating that a control-Y was typed on the terminal.
: 1111    1097 1
: 1112    1098 1 Calling Sequence:
: 1113    1099 1 standard
: 1114    1100 1
: 1115    1101 1 Input Parameters:
: 1116    1102 1 none
: 1117    1103 1
: 1118    1104 1 Implicit Inputs:
: 1119    1105 1 none
: 1120    1106 1
: 1121    1107 1 Output Parameters:
: 1122    1108 1 none
: 1123    1109 1
: 1124    1110 1 Implicit Outputs:
: 1125    1111 1 none
: 1126    1112 1
: 1127    1113 1 Routines Called:
: 1128    1114 1 none
: 1129    1115 1
: 1130    1116 1 Routine Value:
: 1131    1117 1 none
: 1132    1118 1
: 1133    1119 1 Signals:
: 1134    1120 1 none
: 1135    1121 1
: 1136    1122 1 Side Effects:
: 1137    1123 1 A SWAKE will be issued to abort the program.
: 1138    1124 1
: 1139    1125 1 --
: 1140    1126 2 BEGIN
: 1141    1127 2 QUIT;
: 1142    1128 1 END;

```

0000 00000 CNTRLYAST:

| | | | | | |
|-----------|----|-------------|-------|----------------|------|
| | | | .WORD | Save nothing | 1092 |
| 00000000G | 00 | 7E D4 00002 | CLRL | -(SP) | 1126 |
| 00000000G | 00 | 01 FB 00004 | CALLS | #1, SYSSSETAST | |
| 00000000G | 00 | 01 90 00008 | MOV B | #1, WAKEFLAG | |
| | | 7E 7C 00012 | CLR Q | -(SP) | |
| | | 02 FB 00014 | CALLS | #2, SYSSWAKE | |
| | | 04 0001B | RET | | 1128 |

; Routine Size: 28 bytes, Routine Base: \$CODE\$ + 0763

; 1143 1129 1

```
1145      1130 1 ROUTINE CANCEL(BUFFER): NOVALUE =
1146      1131 1 ++
1147      1132 1
1148      1133 1 Functional Description:
1149      1134 1     Cancel I/O's as requested by RSX.
1150      1135 1
1151      1136 1 Calling Sequence:
1152      1137 1     standard
1153      1138 1
1154      1139 1 Input Parameters:
1155      1140 1     BUFFER = address of the link buffer
1156      1141 1
1157      1142 1 Implicit Inputs:
1158      1143 1     IOQUEUE
1159      1144 1     CURRENTIO
1160      1145 1
1161      1146 1 Output Parameters:
1162      1147 1     none
1163      1148 1
1164      1149 1 Implicit Outputs:
1165      1150 1     none
1166      1151 1
1167      1152 1 Routines Called:
1168      1153 1     FREEBUF
1169      1154 1
1170      1155 1 Routine Value:
1171      1156 1     none
1172      1157 1
1173      1158 1 Signals:
1174      1159 1     none
1175      1160 1
1176      1161 1 Side Effects:
1177      1162 1     A completion message is sent to the host.
1178      1163 1
1179      1164 1 --
1180      1165 2
1181      1166 2
1182      1167 2
1183      1168 2
1184      1169 2
1185      1170 3
1186      1171 3
1187      1172 3
1188      1173 4
1189      1174 4
1190      1175 4
1191      1176 3
1192      1177 3
1193      1178 2
1194      1179 3
1195      1180 3
1196      1181 3
1197      1182 4
1198      1183 4
1199      1184 4
1200      1185 4
1201      1186 3

        BEGIN
        MAP BUFFER: REF RTP_BUF;
        LOCAL
        IOBUF: REF RTP_BUF;
        IF .BUFFER[RTP_IDN] EQ[ 255 THEN
          BEGIN
            ! KILL ALL I/O
            SCANCEL (CHAN = .RDWRTCHAN); ! CANCEL CURRENT I/O
            WHILE .IOQUEUE[0] NEQ IOQUEUE DO
              BEGIN
                REMQUE(.IOQUEUE,IOBUF); ! GET NEXT I/O
                FREEBUF(.IOBUF);
              END;
          END;
        ELSE
          BEGIN
            ! KILL ONLY ONE I/O
            IF .CURRENTIO NEQ 0
              AND .CURRENTIO[RTP_IDN] EQL .BUFFER[RTP_IDN] THEN
              BEGIN
                CURRENTIO = 0;
                SCANCEL (CHAN = .RDWRTCHAN);
              END;
          ELSE

```

I 5
16-Sep-1984 02:18:51
14-Sep-1984 13:04:57VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RTPAD.SRC]RSXRT.B32;1Page 43
(16)

```

1202    1187 4
1203    1188 4
1204    1189 5
1205    1190 5
1206    1191 5
1207    1192 6
1208    1193 6
1209    1194 6
1210    1195 5
1211    1196 4
1212    1197 3
1213    1198 2
1214    1199 2
P 1200 2
P 1201 2
P 1202 2
P 1203 2
P 1204 2
P 1205 2
1221    1206 2
1222    1207 1

        BEGIN
        WHILE .IOQUEUE NEQ IOQUEUE DO
            BEGIN
                IOBUF = .IOQUEUE;
                IF .IOBUF[RTP_IDN] EQL .BUFFER[RTP_IDN] THEN
                    BEGIN
                        REMQUE(.IOBUF, IOBUF);
                        FREEBUF(.IOBUF);
                    END;
            END;
        END;

        BUFFER[RTP_FLG] = 0;
        SQIO    (CRAN = .LINKCHAN,
                  FUNC = IOS_WRITEVBLK, ! WRITE TO LINK
                  IOSB = BUFFER[RTP_IOS],
                  ASTADR = LINKWRTDONE,
                  ASTPRM = .BUFFER,
                  P1 = BUFFER[RTP_FNC],
                  P2 = $);

END;

```

| | | | | .EXTRN | SYSSCANCEL | | |
|--|--|----------|--------------------|--------|---------------------|---------------|------|
| | | | 007C 00000 CANCEL: | .WORD | Save R2,R3,R4,R5,R6 | 1130 | |
| | | FF 8F 14 | 00 9E 00002 | MOVAB | RDWRTCHAN, R6 | | |
| | | | 00 9E 00009 | MOVAB | SYSSCANCEL, R5 | | |
| | | | CF 9E 00010 | MOVAB | IOQUEUE, R4 | | |
| | | | AC 00 00015 | MOVL | BUFFER, R2 | 1169 | |
| | | | A2 91 00019 | CMPB | 20(R2), #255 | | |
| | | | 1B 12 0001E | BNEQ | 2\$ | | |
| | | | 66 3C 00020 | MOVZWL | RDWRTCHAN, -(SP) | 1171 | |
| | | | 01 FB 00023 | CALLS | #1, SYSSCANCEL | | |
| | | 50 | 64 9E 00026 | 1\$: | MOVAB | IOQUEUE, R0 | 1172 |
| | | | 64 D1 00029 | CMPL | IOQUEUE, R0 | | |
| | | | 47 13 0002C | BEQL | 4\$ | | |
| | | | 84 0F 0002E | REMQUE | IOQUEUE, IOBUF | 1174 | |
| | | F9 F5 CF | 53 DD 00032 | PUSHL | IOBUF | 1175 | |
| | | | 01 FB 00034 | CALLS | #1, FREEBUF | | |
| | | | EB 11 00039 | BRB | 1\$ | 1172 | |
| | | 50 | F8 A4 00 0038 | 2\$: | MOVL | CURRENTIO, R0 | 1180 |
| | | | 12 13 0003F | BEQL | 3\$ | | |
| | | 14 A2 14 | A0 91 00041 | CMPB | 20(R0), 20(R2) | 1181 | |
| | | | 0B 12 00046 | BNEQ | 3\$ | | |
| | | | A4 D4 00048 | CLRL | CURRENTIO | 1183 | |
| | | | 66 3C 0004B | MOVZWL | RDWRTCHAN, -(SP) | 1184 | |
| | | 7E 65 | 01 FB 0004E | CALLS | #1, SYSSCANCEL | | |
| | | | 22 11 00051 | BRB | 4\$ | 1180 | |
| | | 50 | 64 9E 00053 | 3\$: | MOVAB | IOQUEUE, R0 | 1188 |
| | | | 64 D1 00056 | CMPL | IOQUEUE, R0 | | |
| | | | 1A 13 00059 | BEQL | 4\$ | | |
| | | 53 | 64 00 0005B | MOVL | IOQUEUE, IOBUF | 1190 | |
| | | 50 | 04 AC 00 0005E | MOVL | BUFFER, R0 | 1191 | |
| | | 14 A0 14 | A3 91 00062 | CMPB | 20(IOBUF), 20(R0) | | |
| | | | EA 12 00067 | BNEQ | 3\$ | | |

| | | | | | |
|------|--------------|----|-------------|------------------------|------|
| | 53 | 63 | OF 00069 | REMQUE (IOBUF), IOBUF | 1193 |
| F98B | CF | 53 | DD 0006C | PUSHL IOBUF | 1194 |
| | | 01 | FB 0006E | CALLS #1, FREEBUF | |
| | | DE | 11 00073 | BRB 3\$ | |
| | 50 | 04 | AC DD 00075 | MOVL BUFFER, R0 | 1188 |
| | | 12 | A0 94 00079 | CLRB 18(R0) | 1199 |
| | | | 7E 7C 0007C | CLRQ -(SP) | |
| | | | 7E 7C 0007E | CLRQ -(SP) | |
| | | | 05 DD 00080 | PUSHL #5 | |
| | 10 | A0 | 9F 00082 | PUSHAB 16(R0) | |
| | | 50 | DD 00085 | PUSHL R0 | |
| | FD63 | CF | 9F 00087 | PUSHAB LINKWRTDONE | |
| | 08 | A0 | 9F 0008B | PUSHAB 8(R0) | |
| | | 30 | DD 0008E | PUSHL #48 | |
| | 7E 00000000G | 00 | 3C 00090 | MOVZWL LINKCHAN, -(SP) | |
| | | | 7E D4 00097 | CLRL -(SP) | |
| | 00000000G | 00 | OC FB 00099 | CALLS #12, SYSSQIO | |
| | | | 04 000A0 | RET | 1207 |

; Routine Size: 161 bytes, Routine Base: \$CODE\$ + 077F

```

: 1224      1208 1 ROUTINE MAPMODIFIER(RSXMOD) =
: 1225      1209 1 ++
: 1226      1210 1
: 1227      1211 1 Functional Description:
: 1228      1212 1 Convert RSX function code modifiers to VMS format.
: 1229      1213 1
: 1230      1214 1 Calling Sequence:
: 1231      1215 1 standard
: 1232      1216 1
: 1233      1217 1 Input Parameters:
: 1234      1218 1 RSXMOD = RSX modifiers
: 1235      1219 1
: 1236      1220 1 Implicit Inputs:
: 1237      1221 1 none
: 1238      1222 1
: 1239      1223 1 Output Parameters:
: 1240      1224 1 none
: 1241      1225 1
: 1242      1226 1 Implicit Outputs:
: 1243      1227 1 none
: 1244      1228 1
: 1245      1229 1 Routines Called:
: 1246      1230 1 none
: 1247      1231 1
: 1248      1232 1 Routine Value:
: 1249      1233 1 VMS function code modifier
: 1250      1234 1
: 1251      1235 1 Signals:
: 1252      1236 1 none
: 1253      1237 1
: 1254      1238 1 Side Effects:
: 1255      1239 1 none
: 1256      1240 1
: 1257      1241 1 -- BEGIN
: 1258      1242 2 LOCAL
: 1259      1243 2
: 1260      1244 2 VMSMOD: :
: 1261      1245 2 VMSMOD = IOSM TRMNOECHO;
: 1262      1246 2 IF (.RSXMOD AND RM RNE) NEQ 0 THEN
: 1263      1247 2     VMSMOD = .VMSMOD+IOSM_NOECHO;
: 1264      1248 2 RETURN(.VMSMOD);
: 1265      1249 1 END;

```

0000 00000 MAPMODIFIER:

| | | | | | | | | |
|----|----|------|----|----|------------|--------|------------------|--------|
| 04 | 50 | 1000 | 8F | 3C | 00002 | .WORD | Save nothing | : 1208 |
| | 6C | | 24 | E1 | 00007 | MOVZWL | #4096, VMSMOD | : 1245 |
| | 50 | 40 | A0 | 9E | 0000B | BBC | #36, RSXMOD, 1\$ | : 1246 |
| | | | | 04 | 0000F 1\$: | MOVAB | 64(R0), VMSMOD | : 1247 |
| | | | | | | RET | | : 1249 |

: Routine Size: 16 bytes, Routine Base: \$CODE\$ + 0820

1267 1 ROUTINE ATTACH(BUFFER): NOVALUE =
1268 1 ++
1269 1 Functional Description:
1270 1 Handle the RSX attach and detach functions.
1271 1 Calling Sequence:
1272 1 standard
1273 1 Input Parameters:
1274 1 BUFFER = address of the link buffer
1275 1 Implicit Inputs:
1276 1 CURRENTIO
1277 1 UNSOLPEND
1278 1 Output Parameters:
1279 1 none
1280 1 Implicit Outputs:
1281 1 ATTACHFLAG
1282 1 Routines Called:
1283 1 TERMMBXMSG
1284 1 FREEBUF
1285 1 NEXTIO
1286 1 Routine Value:
1287 1 none
1288 1 Signals:
1289 1 none
1290 1 Side Effects:
1291 1 The request may be queued for later action.
1292 1 If the detach reenables unsolicited input, pending data may be read.
1293 1 --
1294 2 BEGIN
1295 2 MAP BUFFER: REF RTP_BUF;
1296 2 IF .CURRENTIO EQ 0 THEN
1297 3 BEGIN
1298 3 IF .BUFFER[RTP MOD] NEQ RM_DET THEN
1299 3 ATTACHFLAG = 1
1300 3 ELSE
1301 4 BEGIN
1302 4 ATTACHFLAG = 0;
1303 4 IF (.UNSOLPEND NEQ 0) OR (.INDDATA NEQ 0) THEN
1304 5 BEGIN ! DATA ALREADY PENDING
1305 5 TERMMBXDATA[0] = MSGS_TRMUNSOLIC;
1306 5 TERMMBXMSG();
1307 5 END;
1308 6 END:
1309 6 FREEBUF(.BUFFER);
1310 6 NEXTIO(); ! CHECK FOR A PENDING I/O
1311 6 END
1312 7 ELSE
1313 7
1314 7
1315 7
1316 7
1317 7
1318 7
1319 7
1320 7
1321 7
1322 7
1323 7

: 1324

1307 2
1308 1

END; INSQUE(.BUFFER,.IOQUEUE[1]); ! QUEUE IT FOR LATER

| | | | | | | | |
|--|-------|----|----------------|--------------------|---------------------|----------------|------|
| | | | | 0004 00000 ATTACH: | .WORD | Save R2 | 1250 |
| | | 52 | 0000' | CF 9E 00002 | MOVAB | ATTACHFLAG, R2 | |
| | | | 08 | A2 D5 00007 | TSTL | CURRENTIO | 1290 |
| | | | 33 | 12 0000A | BNEQ | 4S | |
| | | 80 | 50 04 | AL D0 0000C | MOVL | BUFFER, R0 | 1292 |
| | | | 8F 11 | A0 91 00010 | CMPB | 17(R0), #128 | |
| | | | | 05 13 00015 | BEQL | 1S | |
| | | | 62 | 01 90 00017 | MOVW | #1, ATTACHFLAG | 1293 |
| | | | | 15 11 0001A | BRB | 3S | |
| | | | | 62 94 0001C | 1\$: CLRBL | ATTACHFLAG | 1296 |
| | | | | 02 A2 95 0001E | TSTB | UNSOLPEND | 1297 |
| | | | | 05 12 00021 | BNEQ | 2S | |
| | | | | 0C A2 D5 00023 | TSTL | INDDATA | |
| | | | | 09 13 00026 | BEQL | 3S | |
| | F4 | A2 | 01 B0 00028 | 2\$: MOVW | #1, TERMMBXDATA | 1299 | |
| | FD98 | CF | 00 FB 0002C | CALLS | #0, TERMMBXMSG | 1300 | |
| | F944 | CF | 04 AC DD 00031 | 3\$: PUSHL | BUFFER | 1303 | |
| | 0000V | CF | 01 FB 00034 | CALLS | #1, FREEBUF | | |
| | | | 00 FB 00039 | CALLS | #0, NEXTIO | 1304 | |
| | | | 04 0003E | RET | | 1290 | |
| | 14 | B2 | 04 BC 0E 0003F | 4\$: INSQUE | aBUFFER, aioQUEUE+4 | 1307 | |
| | | | 04 00044 | PET | | 1308 | |

: Routine Size: 69 bytes. Routine Base: \$CODE\$ + 0830

: 1326 1309 1

1328 1310 1 ROUTINE READSINGLE(BUFFER): NOVALUE =
1329 1311 1 ++
1330 1312 1
1331 1313 1 Functional Description:
1332 1314 1 Enable and disable RSX single character mode.
1333 1315 1
1334 1316 1 Calling Sequence:
1335 1317 1 ;standard
1336 1318 1
1337 1319 1 Input Parameters:
1338 1320 1 BUFFER = address of the link buffer
1339 1321 1
1340 1322 1 Implicit Inputs:
1341 1323 1 CURRENTIO
1342 1324 1 UNSOLPEND
1343 1325 1 SINGLEINPROG
1344 1326 1
1345 1327 1 Output Parameters:
1346 1328 1 none
1347 1329 1
1348 1330 1 Implicit Outputs:
1349 1331 1 SINGLEFLAG
1350 1332 1 UNSOLPEND
1351 1333 1
1352 1334 1 Routines Called:
1353 1335 1 TERMMBXMSG
1354 1336 1 FREEBUF
1355 1337 1
1356 1338 1 Routine Value:
1357 1339 1 none
1358 1340 1
1359 1341 1 Signals:
1360 1342 1 none
1361 1343 1
1362 1344 1 Side Effects:
1363 1345 1 The request may be queued for later action.
1364 1346 1 If data is pending when the mode is enabled, it is read.
1365 1347 1
1366 1348 1 --
1367 1349 2 BEGIN
1368 1350 2 MAP BUFFER: REF RTP_BUF;
1369 1351 2 IF .CURRENTIO EQ 0 THEN
1370 1352 3 BEGIN
1371 1353 3 IF (.BUFFER[RTP_MOD] AND RM_TSC) EQ 0 THEN
1372 1354 4 BEGIN
1373 1355 4 SINGLEFLAG = .BUFFER; ! ENABLE SINGLE CHARACTERS
1374 1356 4 IF .UNSOLPEND NEQ 0 THEN
1375 1357 4 TERMMBXMSG(); ! DATA ALREADY PENDING
1376 1358 4 UNSOLPEND = 0;
1377 1359 4 END
1378 1360 3 ELSE
1379 1361 4 BEGIN ! DISABLE SINGLE CHARACTER MODE
1380 1362 4 FREEBUF(.BUFFER); ! OF NO USE
1381 1363 4 IF .SINGLEINPROG EQ 0 THEN
1382 1364 4 FREEBUF(.SINGLEFLAG); ! NOT CURRENTLY IN USE
1383 1365 4 SINGLEFLAG = 0;
1384 1366 3 END;

```
: 1385      1367 3      NEXTIO();          ! IS ANYTHING ELSE QUEUED
: 1386      1368 3      END
: 1387      1369 2      ELSE
: 1388      1370 2      INSQUE(.BUFFER,.IOQUEUE[1]); ! QUEUE IT FOR LATER
: 1389      1371 1      END;
```

| 0004 00000 READSINGLE: | | | | | | | | |
|------------------------|-------|----|-------|-------|-------------|---------------------|--------------|------|
| | | | | .WORD | Save R2 | | 1310 | |
| 52 | 0000' | CF | 9E | 00002 | MOVAB | SINGLEFLAG, R2 | | |
| | 04 | A2 | D5 | 00007 | TSTL | CURRENTIO | 1351 | |
| | | 38 | 12 | 0000A | BNEQ | 5\$ | | |
| 50 | 04 | AC | D0 | 0000C | MOVL | BUFFER, R0 | 1353 | |
| | 11 | A0 | 95 | 00010 | TSTB | 17(R0) | | |
| | | 13 | 19 | 00013 | BLSS | 2\$ | | |
| 62 | 04 | AC | D0 | 00015 | MOVL | BUFFER, SINGLEFLAG | 1355 | |
| | FE | A2 | 95 | 00019 | TSTB | UNSOLOPEND | 1356 | |
| | | 05 | 13 | 0001C | BEQL | 1\$ | | |
| FD61 | CF | 00 | FB | 0001E | CALLS | #0, TERMMBXMSG | 1357 | |
| | | FE | A2 | 94 | 00023 | 1\$: CLRBL | 1358 | |
| | | 14 | 11 | 00026 | BRB | 4\$ | 1353 | |
| F908 | CF | 04 | AC | DD | 00028 | 2\$: PUSHL | 1362 | |
| | | 01 | FB | 0002B | CALLS | #1, FREEBUF | | |
| | | FD | A2 | 95 | 00030 | TSTB | SINGLEINPROG | 1363 |
| | | 07 | 12 | 00033 | BNEQ | 3\$ | | |
| | | 62 | DD | 00035 | PUSHL | SINGLEFLAG | 1364 | |
| F8FC | CF | 01 | FB | 00037 | CALLS | #1, FREEBUF | | |
| | | 62 | D4 | 0003C | 3\$: CLRL | SINGLEFLAG | 1365 | |
| 0000V | CF | 00 | FB | 0003E | 4\$: CALLS | #0, NEXTIO | 1367 | |
| | | 04 | 00043 | | RET | | 1351 | |
| 10 | B2 | 04 | BC | 00044 | 5\$: INSQUE | @BUFFER, @IOQUEUE+4 | 1370 | |
| | | | | 04 | 00049 | RET | 1371 | |

: Routine Size: 74 bytes, Routine Base: \$CODE\$ + 0875

: 1390 1372 1

```

: 1392 1373 1 ROUTINE ONECHAR(BUFFER): NOVALUE =
: 1393 1374 1 ++
: 1394 1375 1
: 1395 1376 1 Functional Description:
: 1396 1377 1 Handle the completion of a single character mode read.
: 1397 1378 1
: 1398 1379 1 Calling Sequence:
: 1399 1380 1 standard
: 1400 1381 1
: 1401 1382 1 Input Parameters:
: 1402 1383 1 BUFFER = address of the link buffer
: 1403 1384 1
: 1404 1385 1 Implicit Inputs:
: 1405 1386 1 SINGLEFLAG
: 1406 1387 1
: 1407 1388 1 Output Parameters:
: 1408 1389 1 none
: 1409 1390 1
: 1410 1391 1 Implicit Outputs:
: 1411 1392 1 SINGLEINPROG
: 1412 1393 1
: 1413 1394 1 Routines Called:
: 1414 1395 1 QIODONE
: 1415 1396 1 FREEBUF
: 1416 1397 1
: 1417 1398 1 Routine Value:
: 1418 1399 1 none
: 1419 1400 1
: 1420 1401 1 Signals:
: 1421 1402 1 none
: 1422 1403 1
: 1423 1404 1 Side Effects:
: 1424 1405 1 none
: 1425 1406 1
: 1426 1407 1 --
: 1427 1408 2 BEGIN
: 1428 1409 2 LOCAL
: 1429 1410 2 NEWBUF: REF VECTOR;
: 1430 1411 2 MAP BUFFER: REF VECTOR;
: 1431 1412 2 SINGLEINPROG = 0;
: 1432 1413 2 NEWBUF = GETBUF(); ! GET A NEW BUFFER
: 1433 1414 2 NEWBUF[4] = .BUFFER[4];
: 1434 1415 2 NEWBUF[5] = .BUFFER[5];
: 1435 1416 2 NEWBUF[6] = .BUFFER[6];
: 1436 1417 2 QIODONE(.NEWBUF);
: 1437 1418 2 IF .SINGLEFLAG EQL 0 THEN
: 1438 1419 2      FREEBUF(.BUFFER); ! SINGLE CHAR MODE WAS DISABLED
: 1439 1420 1 END;

```

| | | | | |
|------|----|--------------------------------|------------------|--------|
| F8BF | CF | 0000' 0004 00000 ONECHAR: WORD | Save R2 | : 1373 |
| | 52 | 00 94 00002 | CLRB | : 1412 |
| | | 04 FB 00006 | CALLS #0, GETBUF | : 1413 |
| | | AC D0 0000B | MOVL BUFFER, R2 | : 1414 |

| | | | | |
|---------|-------|---------------|-------------------------|--------|
| 10 A0 | 10 A2 | 7D 0000F | MOVQ 16(R2), 16(NEWBUF) | : |
| 18 A0 | 18 A2 | DD 00014 | MOVL 24(R2), 24(NEWBUF) | : 1416 |
| | | 50 DD 00019 | PUSHL NEWBUF | : 1417 |
| FB84 CF | 0000' | 01 FB 0001B | CALLS #1, QIODONE | |
| | | CF D5 00020 | TSTL SINGLEFLAG | : 1418 |
| | | 07 12 00024 | BNEQ 1\$ | |
| F8C1 CF | | 52 DD 00026 | PUSHL R2 | : 1419 |
| | | G1 FB 00028 | CALLS #1, FREEBUF | |
| | | 04 0002D 1\$: | RET | : 1420 |

: Routine Size: 46 bytes, Routine Base: \$CODE\$ + 08BF

```

: 1441 1421 1 ROUTINE TERMINATOR(RSXMOD) =
: 1442 1422 1 ++
: 1443 1423 1
: 1444 1424 1 Functional Description:
: 1445 1425 1 Provide the correct terminator mask for an RSX read operation.
: 1446 1426 1
: 1447 1427 1 Calling Sequence:
: 1448 1428 1 standard
: 1449 1429 1
: 1450 1430 1 Input Parameters:
: 1451 1431 1 RSXMOD = RSX function modifiers
: 1452 1432 1
: 1453 1433 1 Implicit Inputs:
: 1454 1434 1 none
: 1455 1435 1
: 1456 1436 1 Output Parameters:
: 1457 1437 1 none
: 1458 1438 1
: 1459 1439 1 Implicit Outputs:
: 1460 1440 1 none
: 1461 1441 1
: 1462 1442 1 Routines Called:
: 1463 1443 1 none
: 1464 1444 1
: 1465 1445 1 Routine Value:
: 1466 1446 1 address of the descriptor for the terminator mask
: 1467 1447 1
: 1468 1448 1 Signals:
: 1469 1449 1 none
: 1470 1450 1
: 1471 1451 1 Side Effects:
: 1472 1452 1 none
: 1473 1453 1
: 1474 1454 1 --
: 1475 1455 2 BEGIN
: 1476 1456 2 IF (.RSXMOD AND RM RTC) NEQ 0 THEN
: 1477 1457 3 RETURN(STERMDESC) ! TERMINATE ON CONTROL CHARACTERS
: 1478 1458 2 ELSE
: 1479 1459 2 RETURN(NTERMDESC); ! NORMAL TERMINATORS
: 1480 1460 1 END;

```

0000 00000 TERMINATOR:

| | | |
|----|--|---|
| 06 | 6C .WORD Save nothing 50 BBC #35, RSXMOD, 1\$ 50 MOVAB STERMDESC, R0 50 RET | 23 E1 00002 CF 9E 00006 04 0000B 50 0000' CF 9E 0000C 1\$: MOVAB NTERMDESC, R0 04 00011 RET |
|----|--|---|

: Routine Size: 18 bytes. Routine Base: \$CODE\$ + 0BED

1421
1456
1457
1459
1460

: 1482 1461 1 ROUTINE UNSUPPORTED(BUFFER): NOVALUE =
: 1483 1462 1 ++
: 1484 1463 1
: 1485 1464 1 Functional Description:
: 1486 1465 1 Return an error message to the host for unsupported functions.
: 1487 1466 1
: 1488 1467 1 Calling Sequence:
: 1489 1468 1 standard
: 1490 1469 1
: 1491 1470 1 Input Parameters:
: 1492 1471 1 BUFFER = address of the link buffer
: 1493 1472 1
: 1494 1473 1 Implicit Inputs:
: 1495 1474 1 none
: 1496 1475 1
: 1497 1476 1 Output Parameters:
: 1498 1477 1 none
: 1499 1478 1
: 1500 1479 1 Implicit Outputs:
: 1501 1480 1 RETSTATUS
: 1502 1481 1
: 1503 1482 1 Routines Called:
: 1504 1483 1 none
: 1505 1484 1
: 1506 1485 1 Routine Value:
: 1507 1486 1 none
: 1508 1487 1
: 1509 1488 1 Signals:
: 1510 1489 1 none
: 1511 1490 1
: 1512 1491 1 Side Effects:
: 1513 1492 1 If there is an error on the write to the link, a SWAKE is issued to
: 1514 1493 1 abort the program.
: 1515 1494 1
: 1516 1495 1 --
: 1517 1496 2 BEGIN
: 1518 1497 2 MAP BUFFER: REF RTP_BUF;
: 1519 1498 2 RETSTATUS =
P 1499 2 SQIO (CHAN = .LINKCHAN, ! WRITE TO LINK
P 1500 2 FUNC = IOS_WRITEVBLK,
P 1501 2 IOSB = BUFFER[RTP_IOS],
P 1502 2 ASTADR = LINKWRDONE,
P 1503 2 ASTPRM = BUFFER,
P 1504 2 P1 = BUFFER[RTP_FNC],
P 1505 2 P2 = 128);
: 1506 2 IF .RETSTATUS EQL \$SS_ABORT THEN
: 1507 2 RETURN; ! Link gone - mailbox msg will tell why
: 1508 2 QUIT_ON_ERROR;
: 1509 1 END;

0000 00000 UNSUPPORTED:
7E 7C 00002 .WORD CLRQ Save nothing
-(SP)

: 1461
: 1505

| | | | | | | | |
|-----------|----|----|--------------|----|------------|--------------------------|------|
| 7E | 04 | 7E | 80 | 7E | 7C 00004 | CLRQ -(SP) | |
| | | AC | | 8F | 9A 00006 | MOVZBL #128, -(SP) | |
| | | | 04 | 10 | C1 0000A | ADDL3 #16, BUFFER, -(SP) | |
| | | | FC58 | AC | DD 0000F | PUSHL BUFFER | |
| 7E | 04 | AC | | CF | 9F 00012 | PUSHAB LINKWRTDONE | |
| | | | | 08 | C1 00016 | ADDL3 #8, BUFFER, -(SP) | |
| | | | | 30 | DD 0001B | PUSHL #48 | |
| | | | 7E 00000000G | 00 | 3C 0001D | MOVZWL LINKCHAN, -(SP) | |
| 00000000G | 00 | | | 7E | D4 00024 | CLRL -(SP) | |
| 00000000G | 00 | | | 0C | FB 00026 | CALLS #12, SYSSQIO | |
| | | | | 50 | DO 0002D | MOVL R0, RETSTATUS | |
| | | | 2C | 50 | D1 00034 | CMPL R0, #44 | 1506 |
| | | | | 1C | 13 00037 | BEQL 1\$ | |
| | 19 | | | 50 | E8 00C39 | BLBS R0, 1\$ | 1507 |
| 00000000G | 00 | | | 7E | D4 0003C | CLRL -(SP) | |
| 00000000G | 00 | | | 01 | FB 0003E | CALLS #1, SYSSSETAST | |
| | | | | 01 | 90 00045 | MOVB #1, WAKEFLAG | |
| 00000000G | 00 | | | 7E | 7C 0004C | CLRQ -(SP) | |
| | | | | 02 | FB 0004E | CALLS #2, SYSSWAKE | |
| | | | | 04 | 00055 1\$: | RET | 1509 |

: Routine Size: 86 bytes. Routine Base: \$CODE\$ + 08FF

```
: 1532      1510 1 ROUTINE NEXTIO: NOVALUE =
: 1533      1511 1 ++
: 1534      1512 1
: 1535      1513 1 Functional Description:
: 1536      1514 1 Perform the next I/O on the queue.
: 1537      1515 1
: 1538      1516 1 Calling Sequence:
: 1539      1517 1 standard
: 1540      1518 1
: 1541      1519 1 Input Parameters:
: 1542      1520 1 none
: 1543      1521 1
: 1544      1522 1 Implicit Inputs:
: 1545      1523 1 IOQUEUE
: 1546      1524 1 CURRENTIO
: 1547      1525 1
: 1548      1526 1 Output Parameters:
: 1549      1527 1 none
: 1550      1528 1
: 1551      1529 1 Implicit Outputs:
: 1552      1530 1 none
: 1553      1531 1
: 1554      1532 1 Routines Called:
: 1555      1533 1 WRITE
: 1556      1534 1 READ
: 1557      1535 1 READPROMPT
: 1558      1536 1 ATTACH
: 1559      1537 1 READSINGLE
: 1560      1538 1 FREEBUF
: 1561      1539 1
: 1562      1540 1 Routine Value:
: 1563      1541 1 none
: 1564      1542 1
: 1565      1543 1 Signals:
: 1566      1544 1 none
: 1567      1545 1
: 1568      1546 1 Side Effects:
: 1569      1547 1 none
: 1570      1548 1
: 1571      1549 1 --+
: 1572      1550 2 BEGIN
: 1573      1551 2 LOCAL
: 1574      1552 2     NEWIO: REF RTP BUF;
: 1575      1553 2 IF (.IOQUEUE[0] NEQ IOQUEUE) AND (.CURRENTIO EQ 0) THEN
: 1576      1554 3     BEGIN ! TAKE AN I/O OFF THE QUEUE
: 1577      1555 3     REMQUE(.IOQUEUE,NEWIO);
: 1578      1556 3     CASE .NEWIO[RTP_FNC] FROM 3 TO 9 OF
: 1579      1557 3     SET
: 1580      1558 3     [RF_WTD]:      WRITE(.NEWIO);
: 1581      1559 3     [RF_RDD]:      READ(.NEWIO);
: 1582      1560 3     [RF_WRD]:      READPROMPT(.NEWIO);
: 1583      1561 3     [RF_ATT]:      ATTACH(.NEWIO);
: 1584      1562 3     [RF_RSC]:      READSINGLE(.NEWIO);
: 1585      1563 3     [INRANGE]:    FREEBUF(.NEWIO);
: 1586      1564 3     TES:
: 1587      1565 2     END;
: 1588      1566 1     END;
```

| | | | | | | |
|--|-------------------|--------------------------------------|--|----------------------------------|--|------|
| | | | | 0004 00000 NEXTIO: .WORD Save R2 | | 1510 |
| | | 52 0000' CF 9E 00002 | | MOVAB IOQUEUE, R2 | | |
| | | 50 62 9E 00007 | | MOVAB IOQUEUE, R0 | | 1553 |
| | | 50 62 D1 0000A | | CMPL IOQUEUE, R0 | | |
| | | F8 48 13 0000D | | BEQL 8\$ | | |
| | | 50 F8 A2 D5 0000F | | TSTL CURRENTIO | | |
| | | 00 46 12 00012 | | BNEQ 8\$ | | |
| | | 10 B2 0F 00014 | | REMQUE AIOQUEUE, NEWIO | | 1555 |
| | 06 001E 0016 0036 | 000E 00010 1\$: .WORD | | 16(NEWIO), #3, #6 | | 1556 |
| | | 002E 00025 | | 2\$-1\$,- | | |
| | | | | 3\$-1\$,- | | |
| | | | | 4\$-1\$,- | | |
| | | | | 7\$-1\$,- | | |
| | | | | 6\$-1\$,- | | |
| | | | | 7\$-1\$,- | | |
| | | | | 5\$-1\$,- | | |
| | | F915 CF 50 DD 0002B 2\$: PUSHL NEWIO | | CALLS #1, WRITE | | 1558 |
| | | 01 FB 0002D 04 00032 | | RET | | |
| | | F97F CF 50 DD 00033 3\$: PUSHL NEWIO | | CALLS #1, READ | | 1559 |
| | | 01 FB 00035 04 0003A | | RET | | |
| | | FA2C CF 50 DD 0003B 4\$: PUSHL NEWIO | | CALLS #1, READPROMPT | | 1560 |
| | | 01 FB 0003D 04 00042 | | RET | | |
| | | FE91 CF 50 DD 00043 5\$: PUSHL NEWIO | | CALLS #1, ATTACH | | 1561 |
| | | 01 FB 00045 04 0004A | | RET | | |
| | | FECE CF 50 DD 00048 6\$: PUSHL NEWIO | | CALLS #1, READSINGLE | | 1562 |
| | | 01 FB 0004D 04 00052 | | RET | | |
| | | F7FE CF 50 DD 00053 7\$: PUSHL NEWIO | | CALLS #1, FREEBUF | | 1563 |
| | | 01 FB 00055 04 0005A 8\$: RET | | RET | | 1566 |

: Routine Size: 91 bytes. Routine Base: \$CODE\$ + 0955

: 1589 1567 1

```
: 1591      1568 1 ROUTINE LINKMBXMSG: NOVALUE =
: 1592      1569 1 ++
: 1593      1570 1 Functional Description:
: 1594      1571 1 Handle messages received on the link mailbox.
: 1595      1572 1
: 1596      1573 1 Calling Sequence:
: 1597      1574 1 standard
: 1598      1575 1
: 1599      1576 1 Input Parameters:
: 1600      1577 1 none
: 1601      1578 1
: 1602      1579 1 Implicit Inputs:
: 1603      1580 1 none
: 1604      1581 1
: 1605      1582 1 Output Parameters:
: 1606      1583 1 none
: 1607      1584 1
: 1608      1585 1 Implicit Outputs:
: 1609      1586 1 RETSTATUS
: 1610      1587 1
: 1611      1588 1 Routines Called:
: 1612      1589 1 none
: 1613      1590 1
: 1614      1591 1 Routine Value:
: 1615      1592 1 none
: 1616      1593 1
: 1617      1594 1 Signals:
: 1618      1595 1 none
: 1619      1596 1
: 1620      1597 1 Side Effects:
: 1621      1598 1 A new read on the link mailbox may be initiated.
: 1622      1599 1 A SWAKE may be issued to abort the program in case of a link error.
: 1623      1600 1
: 1624      1601 1
: 1625      1602 1 --
: 1626      1603 2 BEGIN
: 1627      1604 2 IF (.LINKMAIL[0] EQL MSG$_DISCON) OR (.LINKMAIL[0] EQL MSG$_ABORT) THEN
: 1628      1605 3   BEGIN ! TIME TO QUIT
: 1629      1606 3     SPUTMSG (MSGVEC = UPLIT(2,REMS_NETDIS,0));
: 1630      1607 3     QUIT;
: 1631      1608 3   END
: 1632      1609 3
: 1633      1610 2 ELSE
: 1634      1611 3   BEGIN ! IGNORE IT
: 1635      1612 3     RETSTATUS =
: 1636      P 1613 3     $QIO    (CHAN = .MAILCHAN,           ! LINK MAILBOX READ
: 1637      P 1614 3     FUNC = IOS_READVBLK,
: 1638      P 1615 3     ASTADR = LINKMBXMSG,
: 1639      P 1616 3     P1 = LINKMAIL,
: 1640      P 1617 3     P2 = 40);
: 1641      1618 3     QUIT_ON_ERROR;
: 1642      1619 2     END;
: 1643      1620 1   END;
```

.PSECT SPLIT\$,NOWRT,NOEXE,2

| | | | | | | | |
|--|--|--------------|-------------|------------------------|------------------|------------|------|
| | | 00000002 | 00044 | P.AAG: | .LUNG 2 | | |
| | | 00000000G | 00048 | .ADDRESS | REMS_NETDIS | | |
| | | 00000000 | 0004C | .LONG 0 | | | |
| | | | | .EXTRN | SYSSPUTMSG | | |
| | | | | .PSECT | \$CODE\$,NOWRT,2 | | |
| | | | | 000C 00000 LINKMBXMSG: | | | |
| | | | | | .WORD | Save R2,R3 | 1568 |
| | | 53 00000000G | CF 9E 00002 | MOVAB | LINKMAIL, R3 | | |
| | | 52 00000000G | 00 9E 00007 | MOVAB | RETSTATUS, R2 | | |
| | | 33 | 63 91 0000E | CMPB | LINKMAIL, #51 | | 1604 |
| | | | 05 13 00011 | BEQL | 1\$ | | |
| | | 30 | 63 91 00013 | CMPB | LINKMAIL, #48 | | |
| | | | 11 12 00016 | BNEQ | 2\$ | | |
| | | | 7E 7C 00018 | 1\$: | CLRQ -(SP) | | 1606 |
| | | | 7E D4 0001A | CLRL | -(SP) | | |
| | | 00000000G 00 | CF 9F 0001C | PUSHAB | P.AAG | | |
| | | | 04 FB 00020 | CALLS | #4, SYSSPUTMSG | | |
| | | | 26 11 00027 | BRB | 3\$ | | |
| | | | 7E 7C 00029 | 2\$: | CLRQ -(SP) | | 1617 |
| | | | 7E 7C 0002B | CLRQ | -(SP) | | |
| | | | 28 DD 0002D | PUSHL | #40 | | |
| | | | 53 DD 0002F | PUSHL | R3 | | |
| | | | 7E D4 00031 | CLRL | -(SP) | | |
| | | CA | AF 9F 00033 | PUSHAB | LINKMBXMSG | | |
| | | 7E 00000000G | 31 7D 00036 | MOVQ | #49, -(SP) | | |
| | | | 00 3C 00039 | MOVZWL | MAILCHAN, -(SP) | | |
| | | 00000000G 00 | 7E D4 00040 | CLRL | -(SP) | | |
| | | | 0C FB 00042 | CALLS | #12, SYSSQIO | | |
| | | 62 | 50 D0 00049 | MOVL | R0, RETSTATUS | | |
| | | 19 | 62 E8 0004C | BLBS | RETSTATUS, 4\$ | | |
| | | 00000000G 00 | 7E D4 0004F | 3\$: | CLRL | -(SP) | |
| | | 00000000G 00 | 01 FB 00051 | CALLS | #1, SYSSSETAST | | |
| | | 00000000G 00 | 01 90 00058 | MOVB | #1, WAKEFLAG | | |
| | | 00000000G 00 | 7E 7C 0005F | CLRQ | -(SP) | | |
| | | | 02 FB 00061 | CALLS | #2, SYSSWAKE | | |
| | | | 04 00068 | 4\$: | RET | | 1620 |

: Routine Size: 105 bytes. Routine Base: \$CODE\$ + 0980

```
1645      1621 1 ROUTINE INDREAD =
1646      1622 1 ++
1647      1623 1
1648      1624 1 Functional Description:
1649      1625 1     Read a record from an indirect command file.
1650      1626 1
1651      1627 1 Calling Sequence:
1652      1628 1     standard
1653      1629 1
1654      1630 1 Input Parameters:
1655      1631 1     none
1656      1632 1
1657      1633 1 Implicit inputs
1658      1634 1     INDDATA
1659      1635 1     INDFLAG
1660      1636 1     SYSINRAB
1661      1637 1     SYSINFAB
1662      1638 1
1663      1639 1
1664      1640 1 Output Parameters:
1665      1641 1     none
1666      1642 1
1667      1643 1 Implicit Outputs:
1668      1644 1     SYSINRAB
1669      1645 1
1670      1646 1 Routines Called:
1671      1647 1     $GET
1672      1648 1     $CLOSE
1673      1649 1     FREEBUF
1674      1650 1
1675      1651 1 Routine Value:
1676      1652 1     Status of the $GET
1677      1653 1
1678      1654 1 Signals:
1679      1655 1     none
1680      1656 1
1681      1657 1 Side Effects:
1682      1658 1     If an EOF is read, the indirect command file is closed.
1683      1659 1
1684      1660 1 --.
1685      1661 2 BEGIN
1686      1662 2 RETSTATUS =
1687      1663 2 $GET (RAB = SYSINRAB); ! READ A RECORD
1688      1664 2 IF .RETSTATUS EQL RMSS_EOF THEN
1689      1665 3     BEGIN ! END OF FILE
1690      1666 3     $CLOSE (FAB = SYSINFAB); ! CLOSE THE COMMAND FILE
1691      1667 3     FREEBUF(.INDDATA); ! GET RID OF THE BUFFER
1692      1668 3     INDDATA = 0; ! NO MORE DATA
1693      1669 3     INDFLAG = 0; ! NO MORE FILE
1694      1670 3     END
1695      1671 2 ELSE
1696      1672 3     BEGIN
1697      1673 3     IF (.RETSTATUS AND 1) EQL 0 THEN RETURN .RETSTATUS; ! ERROR
1698      1674 3     (.INDDATA+26+.SYSINRAB[RABSW_RSZ])<0,8> = %X'0D'; ! ADD TERMINATOR
1699      1675 3     INDDATA[RTP_IOC] = .SYSINRAB[RABSW_RSZ]; ! RECORD SIZE
1700      1676 3     INDDATA[RTP_IOS] = .RETSTATUS; ! STATUS FROM THE $GET
1701      1677 2     END;
```

: 1702

1678 2
1679 1RETURN .RETSTATUS;
END;

| | | | | | .EXTRN SYSSGET, SYSCLOSE | |
|-----------|--------------|--------------|-------------|---------------------------|--------------------------|------|
| | | | | 001C 00000 INDREAD: .WORD | Save R2,R3,R4 | 1621 |
| | | 54 00000000G | CF 9E 00002 | MOVAB INDDATA, R4 | | |
| | | 53 00000000G | 00 9E 00007 | MOVAB RETSTATUS, R3 | | 1663 |
| 00000000G | 00 | 00000000G | 00 9F 0000E | PUSHAB SYSINRAB | | |
| | | | 01 FB 00014 | CALLS #1, SYSSGET | | |
| | 63 | | 50 D0 0001B | MOVL R0, RETSTATUS | | |
| 0001827A | 8F | | 52 D0 0001E | MOVL RETSTATUS, R2 | | 1664 |
| | | | 52 D1 00021 | CPL R2, #98938 | | |
| | | | 1E 12 00028 | BNEQ 1S | | |
| 00000000G | 00 | 00000000G | 00 9F 0002A | PUSHAB SYSINFAB | | 1666 |
| | | | 01 FB 00030 | CALLS #1, SYSCLOSE | | |
| F756 | CF | | 64 DD 00037 | PUSHL INDDATA | | 1667 |
| | | | 01 FB 00039 | CALLS #1, FREEBUF | | |
| | | 00000000G | 64 D4 0003E | CLRL INDDATA | | 1668 |
| | | | 00 94 00040 | CLRB INDFLAG | | 1669 |
| | 04 | | 1E 11 00046 | BRB 3S | | 1664 |
| | 50 | | 52 E8 00048 | 1S: BLBS R2, 2\$ | | 1673 |
| | | | 52 D0 0004B | MOVL R2, R0 | | |
| | | | 04 0004E | RET | | |
| | 50 | | 64 D0 0004F | 2S: MOVL INDDATA, R0 | | 1674 |
| 1A A140 | 51 00000000G | | 00 3C 00052 | MOVZWL SYSINRAB+34, R1 | | |
| 0A A0 | | | 00 90 00059 | MOVB #13, 26(R1)[R0] | | |
| 08 A0 | | | 51 B0 0005E | MOVW R1, 10(R0) | | 1675 |
| | | | 52 B0 00062 | MOVW R2, 8(R0) | | 1676 |
| | 50 | | 63 D0 00066 | 3S: MOVL RETSTATUS, R0 | | 1678 |
| | | | 04 00069 | RET | | 1679 |

: Routine Size: 106 bytes, Routine Base: \$CODE\$ + 0A19

```
1705    1680 1 ROUTINE GETTERMCHAR(BUFFER): NOVALUE =
1706    1681 1 ++
1707    1682 1
1708    1683 1 Functional Description:
1709    1684 1     Return the terminal characteristics
1710    1685 1
1711    1686 1 Calling Sequence:
1712    1687 1     standard
1713    1688 1
1714    1689 1 Input Parameters:
1715    1690 1     BUFFER = address of buffer from link
1716    1691 1
1717    1692 1 Implicit Inputs:
1718    1693 1     none
1719    1694 1
1720    1695 1 Output Parameters:
1721    1696 1     none
1722    1697 1
1723    1698 1 Implicit Outputs:
1724    1699 1     none
1725    1700 1
1726    1701 1 Routines Called:
1727    1702 1     none
1728    1703 1
1729    1704 1 Routine Value:
1730    1705 1     none
1731    1706 1
1732    1707 1 Signals:
1733    1708 1     none
1734    1709 1
1735    1710 1 Side Effects:
1736    1711 1     none
1737    1712 1
1738    1713 1 --
1739    1714 2 BEGIN
1740    1715 2 LOCAL
1741    1716 2     CHARPTR : REF VECTOR[,BYTE],
1742    1717 2             CHARBUF : VECTOR[3];
1743    1718 2
1744    1719 2     MAP
1745    1720 2     BIND
1746    1721 2     BUFFER : REF RTP_BUF;
1747    1722 2
1748    1723 2     TERMTYPE = CHARBUF+1 : BYTE,
1749    1724 2             TERMWIDTH = CHARBUF+2 : WORD,
1750    1725 2             TERMCHAR = CHARBUF[1] : BLOCK[,BYTE],
1751    1726 2             TERMLENGTH = CHARBUF[1]+3 : BYTE,
1752    1727 2             TERMCHAR2 = CHARBUF[3] : BLOCK[,BYTE];
1753    P 1728 2
1754    P 1729 2     RETSTATUS =
1755    P 1730 2             $QIOW (CHAN = .CNTRLCHAN,
1756    1731 2                 FUNC = IOS_SENSEMODE,
1757    1732 2                 P1 = CHARBUF,
1758    1733 2                 P2 = 12);
1759    1734 2     QUIT ON_ERROR:
1760    1735 2             CHARPTR = BUFFER[RTP DAT];      ! POINT TO THE CHARACTERISTICS LIST
1761    1736 3             UNTIL .CHARPTR[0] EQ[ 0
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
20100
20101
20102
20103
20104
20105
20106
20107
20108
20109
20110
20111
20112
20113
20114
20115
20116
20117
20118
20119
20120
20121
20122
20123
20124
20125
20126
20127
20128
20129
20130
20131
20132
20133
20134
20135
20136
20137
20138
20139
20140
20141
20142
20143
20144
20145
20146
20147
20148
20149
20150
20151
20152
20153
20154
20155
20156
20157
20158
20159
20160
20161
20162
20163
20164
20165
20166
20167
20168
20169
20170
20171
20172
20173
20174
20175
20176
20177
20178
20179
20180
20181
20182
20183
20184
20185
20186
20187
20188
20189
20190
20191
20192
20193
20194
20195
20196
20197
20198
20199
20200
20201
20202
20203
20204
20205
20206
20207
20208
20209
202010
202011
202012
202013
202014
202015
202016
202017
202018
202019
202020
202021
202022
202023
202024
202025
202026
202027
202028
202029
202030
202031
202032
202033
202034
202035
202036
202037
202038
202039
202040
202041
202042
202043
202044
202045
202046
202047
202048
202049
202050
202051
202052
202053
202054
202055
202056
202057
202058
202059
202060
202061
202062
202063
202064
202065
202066
202067
202068
202069
202070
202071
202072
202073
202074
202075
202076
202077
202078
202079
202080
202081
202082
202083
202084
202085
202086
202087
202088
202089
202090
202091
202092
202093
202094
202095
202096
202097
202098
202099
2020100
2020101
2020102
2020103
2020104
2020105
2020106
2020107
2020108
2020109
2020110
2020111
2020112
2020113
2020114
2020115
2020116
2020117
2020118
2020119
2020120
2020121
2020122
2020123
2020124
2020125
2020126
2020127
2020128
2020129
2020130
2020131
2020132
2020133
2020134
2020135
2020136
2020137
2020138
2020139
2020140
2020141
2020142
2020143
2020144
2020145
2020146
2020147
2020148
2020149
2020150
2020151
2020152
2020153
2020154
2020155
2020156
2020157
2020158
2020159
2020160
2020161
2020162
2020163
2020164
2020165
2020166
2020167
2020168
2020169
2020170
2020171
2020172
2020173
2020174
2020175
2020176
2020177
2020178
2020179
2020180
2020181
2020182
2020183
2020184
2020185
2020186
2020187
2020188
2020189
2020190
2020191
2020192
2020193
2020194
2020195
2020196
2020197
2020198
2020199
2020200
2020201
2020202
2020203
2020204
2020205
2020206
2020207
2020208
2020209
20202010
20202011
20202012
20202013
20202014
20202015
20202016
20202017
20202018
20202019
20202020
20202021
20202022
20202023
20202024
20202025
20202026
20202027
20202028
20202029
202020210
202020211
202020212
202020213
202020214
202020215
202020216
202020217
202020218
202020219
202020220
202020221
202020222
202020223
202020224
202020225
202020226
202020227
202020228
202020229
202020230
202020231
202020232
202020233
202020234
202020235
202020236
202020237
202020238
202020239
202020240
202020241
202020242
202020243
202020244
202020245
202020246
202020247
202020248
202020249
202020250
202020251
202020252
202020253
202020254
202020255
202020256
202020257
202020258
202020259
202020260
202020261
202020262
202020263
202020264
202020265
202020266
202020267
202020268
202020269
202020270
202020271
202020272
202020273
202020274
202020275
202020276
202020277
202020278
202020279
202020280
202020281
202020282
202020283
202020284
202020285
202020286
202020287
202020288
202020289
202020290
202020291
202020292
202020293
202020294
202020295
202020296
202020297
202020298
202020299
2020202100
2020202101
2020202102
2020202103
2020202104
2020202105
2020202106
2020202107
2020202108
2020202109
2020202110
2020202111
2020202112
2020202113
2020202114
2020202115
2020202116
2020202117
2020202118
2020202119
2020202120
2020202121
2020202122
2020202123
2020202124
2020202125
2020202126
2020202127
2020202128
2020202129
2020202130
2020202131
2020202132
2020202133
2020202134
2020202135
2020202136
2020202137
2020202138
2020202139
2020202140
2020202141
2020202142
2020202143
2020202144
2020202145
2020202146
2020202147
2020202148
2020202149
2020202150
2020202151
2020202152
2020202153
2020202154
2020202155
2020202156
2020202157
2020202158
2020202159
2020202160
2020202161
2020202162
2020202163
2020202164
2020202165
2020202166
2020202167
2020202168
2020202169
2020202170
2020202171
2020202172
2020202173
2020202174
2020202175
2020202176
2020202177
2020202178
2020202179
2020202180
2020202181
2020202182
2020202183
2020202184
2020202185
2020202186
2020202187
2020202188
2020202189
2020202190
2020202191
2020202192
2020202193
2020202194
2020202195
2020202196
2020202197
2020202198
2020202199
2020202200
2020202201
2020202202
2020202203
2020202204
2020202205
2020202206
2020202207
2020202208
2020202209
2020202210
2020202211
2020202212
2020202213
2020202214
2020202215
2020202216
2020202217
2020202218
2020202219
2020202220
2020202221
2020202222
2020202223
2020202224
2020202225
2020202226
2020202227
2020202228
2020202229
2020202230
2020202231
2020202232
2020202233
2020202234
2020202235
2020202236
2020202237
2020202238
2020202239
2020202240
2020202241
2020202242
2020202243
2020202244
2020
```

```

: 1762      1737 3      CASE .CHARPTR[0] FROM 0 TO RC_MAX OF
: 1763      1738 3      SET
: 1764      1739 3      [RC_HHT]: CHARPTR[1] = .TERMCHAR[TT$V_MECHTAB];
: 1765      1740 3      [RC_NEC]: CHARPTR[1] = .TERMCHAR[TT$V_NOECHO];
: 1766      1741 3      [RC_TTP]: SELECTONE .TERMTYPE OF
: 1767      1742 3      SET
: 1768      1743 3      [DT$_VT100]: CHARPTR[1] = 13;
: 1769      1744 3      [DT$_VT52]: CHARPTR[1] = 9;
: 1770      1745 3      [OTHERWISE]: ;
: 1771      1746 3      TES;
: 1772      1747 3
: 1773      1748 3
: 1774      1749 3
: 1775      1750 3
: 1776      1751 3
: 1777      1752 3
: 1778      1753 3      [RC_SCP]: CHARPTR[1] = .TERMCHAR[TT$V_SCOPE];
: 1779      1754 3      [RC_BIN]: CHARPTR[1] = .TERMCHAR[TT$V_PASSALL];
: 1780      1755 3      [RC_TPL]: CHARPTR[1] = .TERMLENGTH;
: 1781      1756 3      [INRANGE]: ;
: 1782      1757 3      [OUTRANGE]: ;
: 1783      1758 3      TES;
: 1784      1759 3
: 1785      1760 3
: 1786      1761 3      CHARPTR = .CHARPTR + 2;
: 1787      1762 2      END;
: 1788      1763 2      BUFFER[RTP_STS] = RS_SFC; ! GOOD STATUS
: 1789      1764 2      RETSTATUS =
: P 1765 2      $Q10 (CHAN = .LINKCHAN,
: 1790          FUNC = IOS_WRITEVBLK, ! WRITE TO LINK
: P 1766 2          IOSB = BUFFER[RTP_IOS],
: 1791          ASTADR = LINKWRDONE,
: P 1767 2          ASTPRM = .BUFFER,
: 1792          P1 = BUFFER[RTP_FNC],
: 1793          P2 = (.CHARPTR + 2 - BUFFER[RTP_FNC]));
: 1794          IF .RETSTATUS EQL SSS_ABORT THEN
: 1795              RETURN; ! LINK GONE - MAILBOX MESSAGE WILL TELL WHY
: 1796          QUIT_ON_ERROR;
: 1797          END;
: 1798      1773 2
: 1799      1774 2
: 1800      1775 1

```

| 000C 00000 GETTERMCHAR: | | | | | |
|-------------------------|-----------|----------------|--------|------------------|------|
| | | | .WORD | Save R2,R3 | 1680 |
| 53 | 00000000G | 00 9E 00002 | MOVAB | RETSTATUS, R3 | |
| SE | | 0C C2 00009 | SUBL2 | #12, SP | 1731 |
| | | 7E 7C 0000C | CLRQ | -(SP) | |
| | | 7E 7C 0000E | CLRQ | -(SP) | |
| | | 0C DD 00010 | PUSHL | #12 | |
| | | 14 AE 9F 00012 | PUSHAB | CHARBUF | |
| | | 7E 7C 00015 | CLRQ | -(SP) | |
| 7E | 00000000G | 27 7D 00017 | MOVO | #39, -(SP) | |
| | | 00 3C 0001A | MOVZWL | CNTRLCHAN, -(SP) | |
| | | 7E D4 00021 | CLRL | -(SP) | |
| 00000000G | 00 | 0C FB 00023 | CALLS | #12, SYSSQIOW | |

| | | | | | | | | |
|----|-----------|-------------|-------------|----------|---------|-----------------|-------------------------|--------|
| 01 | A0 | 07 | AE | 90 000C2 | 12\$: | MOV B | TERM LENGTH, 1(CHARPTR) | : 1757 |
| | 50 | 02 | C0 000C7 | 13\$: | ADD L2 | #2, CHARPTR | : 1761 | |
| | | 13 | FF 6E | 31 000CA | | BRW | 2\$ | : 1734 |
| | | | A1 | 94 000CD | 14\$: | CLR B | 19(R1) | : 1763 |
| | | | 7E | 7C 000D0 | | CLR Q | -(SP) | : 1771 |
| | | | 7E | 7C 000D2 | | CLR Q | -(SP) | |
| | 52 | 10 | A1 | 9E 000D4 | | MOV AB | 16(R1), R2 | |
| | 50 | 52 | C2 000D8 | | SUBL 2 | R2, R0 | | |
| | | 02 | A0 9F 000DB | | PUSH AB | 2(R0) | | |
| | | 10 | A1 9F 000DE | | PUSH AB | 16(R1) | | |
| | | | 51 DD 000E1 | | PUSHL | R1 | | |
| | | FA03 | CF 9F 000E3 | | PUSH AB | LINKWRDONE | | |
| | | 08 | A1 9F 000E7 | | PUSH AB | 8(R1) | | |
| | | | 30 DD 000EA | | PUSHL | #48 | | |
| | | 7E 0000000G | 00 3C 000EC | | MOV ZWL | LINKCHAN, -(SP) | | |
| | 00000000G | 00 | 7E D4 000F3 | | CLRL | -(SP) | | |
| | 63 | 0C FB 000F5 | | | CALLS | #12, SYSSQIO | | |
| | 2C | 50 D0 000FC | | | MOVL | R0, RETSTATUS | | |
| | | 50 D1 000FF | | | CMPL | R0, #44 | 1772 | |
| | | 1C 13 00102 | | | BEQL | 16\$ | | |
| | 19 | 50 E8 00104 | | | BLBS | R0, 16\$ | 1773 | |
| | | 7E D4 00107 | 15\$: | | CLRL | -(SP) | | |
| | 00000000G | 00 | 01 FB 00109 | | CALLS | #1, SYSSSETAST | | |
| | 00000000G | 00 | 01 90 00110 | | MOVB | #1, WAKEFLAG | | |
| | 00000000G | 00 | 7E 7C 00117 | | CLR Q | -(SP) | | |
| | | 02 FB 00119 | | | CALLS | #2, SYSSWAKE | | |
| | | 04 00120 | 16\$: | | RET | | 1775 | |

: Routine Size: 289 bytes, Routine Base: \$CODE\$ + 0A83

```
: 1801    1776 1
: 1802    1777 1
: 1803    1778 1
: 1804    1779 0
```

END
ELUDOM

PSECT SUMMARY

| Name | Bytes | Attributes |
|----------|-------|--|
| \$OWNS | 164 | NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2) |
| SPLITS | 80 | NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2) |
| PROTOTBL | 6 | NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(0) |
| \$CODES | 2980 | NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2) |

Library Statistics

| File | ----- Symbols ----- | | | Pages Mapped | Processing Time |
|------|---------------------|--------|---------|--------------|-----------------|
| | Total | Loaded | Percent | | |

RSXRT
V04-000

E 7
16-Sep-1984 02:18:51
14-Sep-1984 13:04:57 VAX-11 BLiss-32 v4.0-742
DISK\$VMSMASTER:[RTPAD.SRC]RSXRT.B32;1 Page 65 (26)

:\$255\$DUA28:[SYSLIB]LIB.L32;1
:-\$255\$DUA28:[SYSLIB]CLIMAC.L32;1

18619 14 45 2 0 14 1000 9 00:01.4
00:00.0

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:RSXRT/OBJ=OBJ\$:RSXRT MSRC\$:RSXRT/UPDATE=(ENH\$:RSXRT)

: Size: 2980 code + 250 data bytes
: Run Time: 00:37.4
: Elapsed Time: 02:35.6
: Lines/CPU Min: 2850
: Lexemes/CPU-Min: 36583
: Memory Used: 222 pages
: Compilation Complete

0334 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

